AÑO II - NUM. 47

MICROHOBBY

SEMANAL

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR

AÑO II - N.º 47

125 PTS.

Canarias 135 ptas.

EDITA HOBBY PRESS, S.A.

SONIMAG'85

UNA CITA ANUAL CON LA INFORMATICA

NUEVO

"NODES OF YESOD"

UN PASEO POR LA LUNA CON CHARLIE

CODIGO MAQUINA

INSTRUCCIONES DE CARGA

UTILIDADES

GRAFICOS DE "TORTUGA" ACERCATE AL LOGO

PONCE

Director Editorial
José I. Gómez-Centurión
Director Ejecutivo
Domingo Gómez
Subdirector
Gabriel Nieto
Redactor Jefe
Africa Pérez Tolosa
Diseño
Rosa María Capitel
Redacción
José María Díaz, Miguel Angel Hijosa, Fco. Javier Martín
Secretaria Redacción
Carmen Santamaría
Colaboradores
Jesús Alonso, Primitivo de Francisco, Rafael Prades, Miguel Sepúlveda, Sergio Martínez y J. M. Lazo
Fotografía
Javier Martínez, Carlos Candel
Portada
José María Ponce

Dibujos
J.R. Ballesteros, A. Perera, F.L. Frontán, Pejo, J.M. López Moreno, J. Igual, J.A. Calvo, Lóriga, Luisma, J. Olivares.

Edita
HOBBY PRESS, S.A.

Presidente
María Andrino
Consejero Delegado
José I. Gómez-Centurión

Jefe de Publicidad
Marisa Esteban
Publicidad Barcelona
José Galán Cortés,
Tels.: 303 10 22 - 313 71 76.

Secretaria de Dirección
Marisa Cogorro

Suscripciones
M.ª Rosa González
M.ª del Mar Calzada

Redacción, Administración y Publicidad
La Granja, n.º 8
Polígono Industrial de Alcobendas
Tel.: 654 32 11
Telex: 49480 HOPR

Dto. Circulación
Carlos Peropadre

Distribución
Coedis, S.A. Valencia, 245
Barcelona

Imprime
Rotedic, S.A.
Carretera de Irún, Km. 12,450
Tel.: 734 15 00
Fotocomposición
Espacio y Punto, S.A.
Paseo de la Castellana, 268
Fotomecánica
Grof
Ezequiel Solana, 16
Depósito Legal:
M-36.598-1984

Representante para Argentina, Chile, Uruguay y Paraguay, Cía. Americana de Ediciones, S.R.L. Sud América, 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina).

Solicitado control
OJD

MICROHOBBY ESTA SEMANA

AÑO II. N.º 47. 8 al 14 de octubre de 1985
125 ptas. (Sobretasa Canarias 10 ptas.)



«NODES OF YESOD»

Las aventuras de Charlie el astronauta. *Página 12.*

PREMIADOS HOBBY-SUERTE

CELIO NIETO TORRALBA. Olmos, 28, 2.º Aranjuez (MADRID).
Cinta de programas (5.ª Cat.)
FERNANDO LOBEZ FRAGA. San Pascual, 14. Alcaniz (TERUEL).
Cinta de programas (5.ª Cat.)
MANUEL MAZUELOS ROJA. Paulo Oroso, 5. (SEVILLA).
Cinta de programas (5.ª Cat.)
JOSE GARCIA POZUELO. Avda. Reyes Católicos, 29. Getafe (MADRID).
Cinta de programas (5.ª Cat.)
MARGARITA BAJO DE LA FUENTE. Torras y Bajes, 4. Cornellá (BARCELONA).
Un Joytick con su Interface (3.ª Cat.)
VICTOR MANUEL VAZQUEZ LOPEZ. Avda. de Vigo, 150, 1.º D. Ferrol (CORUÑA).
Cinta de programas (5.ª Cat.)
DAVID BRAVO QUERIDO. Zamora, 2, 3.º. Torrejón de Ardoz (MADRID).
Cinta de programas (5.ª Cat.)
JUAN CRESPO SANCHEZ. Guangell, 119-121, 1.º. (BARCELONA).
Cinta de programas (5.ª Cat.)
JUAN IGNACIO GOMEZ MORENO. Escurzen, 2. 1.º D. (BILBAO).
Un Spectrum 48K (1.ª Cat.)
ANTONIO PEREZ SANZ. Sta. Emilia de Rodar, 1. (CORDOBA).
Cinta de programas (5.ª Cat.)
ALBERTO ANGEL SANCHEZ CASADO. Vélez, 7, 6.º C. Puertollano (CIUDAD REAL).
Cinta de programas (5.ª Cat.)
GERMAN GUTIERREZ HIERRO. Avd. Madariaga, 83, 1.º E. (BILBAO).
Una Impresora GP-50 (2.ª Cat.)
CIEL ROGRIGUEZ. Avda. Roma, 7, 2.º. Lisboa (PORTUGAL).
Suscripción a Microhobby Semanal por un año (4.ª Cat.)
IGNACIO BARRIOS MONTALVO. Avda. Cardenal Herrera Oria, 169. (MADRID).
Un Joystick con su Interface (3.ª Cat.)
JESUS ANGEL NUÑEZ PINAR. Camino Viejo de Villaverde 12, 6.º A. (MADRID).
Suscripción a Microhobby Semanal por un año (4.ª Cat.)
ANDRES CIRUELA MARTINEZ. Abtao, 40. (MADRID).
Suscripción a Microhobby Semanal por un año (4.ª Cat.)
OSCAR PASCUALERA ARGUELLE. Joaquín Beunza, 56. (BARCELONA).
Cinta de programas (5.ª Cat.)
VICENTE JESUS MIRO PLATA. Partida Canaster, 339. San Vicente de Raspei (ALICANTE).
Cinta de programas (5.ª Cat.)
MARISOL SALAMIRO SANCHEZ-GABRIEL. Rafael Salgado, 23. (SEVILLA).
Cinta de programas (5.ª Cat.)
MIGUEL FERNANDEZ-PALACIOS GORDON. Blasco de Garay, 14. (MADRID).
Cinta de programas (5.ª Cat.)
LUIS SANCHO CROUC. San Luis, 60. (BARCELONA).
Cinta de programas (5.ª Cat.)



Del 23 al 29 de septiembre pasado ha tenido lugar en Barcelona el 23 Salón Internacional de la Imagen, el Sonido y la Electrónica, SONIMAG 85.

Televisión, vídeo, Alta Fidelidad, instrumentos musicales y ordenadores, han conformado la exposición de este año que, desgraciadamente, ha contado con pocas novedades en cuanto al capítulo que a nosotros nos compete.

Microhobby estuvo allí para dar cuenta puntual de ello a nuestros lectores.



El stand de Investrónica, uno de los más solicitados ante la presentación del 128K.



Pin-Sof presentó su gran novedad: el digitalizador de imágenes para Spectrum.

La guerra del 128

Las tres compañías de mayor envergadura en fabricación de ordenadores personales domésticos, SINCLAIR, COMMODORE y AMSTRAD presentaban sus respectivos modelos de 128K, si bien es verdad que a la hora de redactar estas líneas ninguno de los tres está todavía disponible en el mercado.

Esta guerra endiablada por lanzar sus respectivos modelos antes que los otros, con mayores prestaciones y a un precio más bajo ha hecho que al SPECTRUM 128K se le conozca con el nombre clave de DERBY, debido a la desenfrenada carrera emprendida para no quedarse atrás en el cada día más competitivo mundo de la informática.

La estrella fue, sin embargo, el 128 de SINCLAIR por dos motivos principalmente: fue presentado como novedad mundial mientras que los otros ya lo fueron en la pasada Personal Computer Show y nunca antes se había dado la circunstancia de ser España el lugar elegido para presentar primicias de este tipo. Esto se debe, como ya comentábamos la pasada semana, por haber sido desarrollado conjuntamente por Sinclair e Investrónica.

Digitalizador de Imágenes

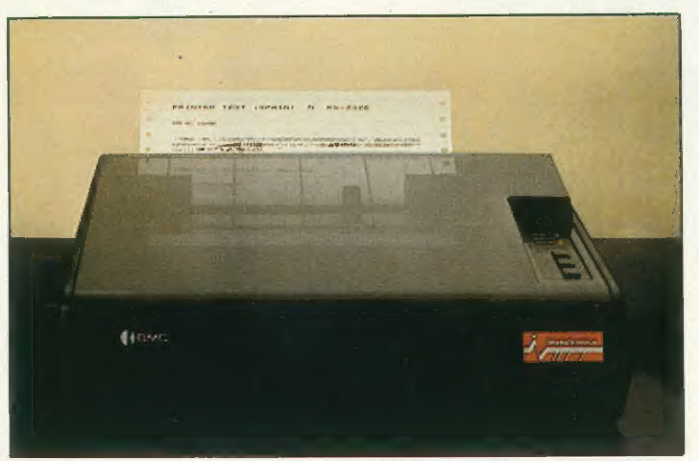
Otra de las novedades que pudimos observar, si bien hasta el momento a nivel de prototipo, fue un DIGITALIZADOR de IMAGENES para Spectrum, presentado por PIN SOFT.

El invento en cuestión es una especie de lápiz óptico que se encuentra adosado a un tambor giratorio. En este último se coloca la fotografía o dibujo que queremos digitalizar. La totalidad de la imagen se consigue explorar gracias a que el tambor gira simultáneamente al desplazamiento lateral del sensor. Tras aproximadamente un minuto, la imagen estará almacenada en memoria con la resolución que permite el Spectrum y, lógicamente, en blanco y negro, ya que el sistema se basa en variaciones de la intensidad luminosa, no cromáticas.

Con un software adicional incluido, se posibilita el manipulado de la imagen, siluetados, inversiones, tramados, etc., al tiempo que permite sacar por impresora copias de las imágenes realizadas, al tamaño deseado y del dibujo completo o porciones del mismo.

El precio, rondará las 35.000 pesetas.

En esta misma línea están trabajando en un software auxiliar que permita imprimir directamente la imagen según es digitalizada por el aparato. Tiene la ventaja de conseguir mayor resolución (tanta como tenga la impresora) al no necesitar un almacenamiento temporal en la memoria del ordenador.



La BX 1000: El 128K ya tiene complementos.

IMPRESORA BX 1000

Investrónica sigue apostando fuerte por el QL en lo que se refiere al mercado de Ordenadores Personales/Profesionales para Gestión. En la idea de potenciar, tanto éste como su nuevo producto, el Spectrum 128K, lanza ahora al mercado una nueva impresora que, cumpliendo la nueva normativa sobre homologación de ordenadores al mercado hispánico, está específicamente diseñada para imprimir todos los caracteres castellanos, acentos, etc.

La BX 1000, pues de ella se trata, es una impresora de bajo coste que tiene el juego de caracteres ASCII completo, pudiendo trabajar además en modo gráfico.

Entre las características técnicas podemos destacar:

- Impresión por impacto de matriz de puntos.
- Escritura a 4Ø, 71, 8Ø y 142 columnas.
- Recorrido bidireccional.
- Caracteres normales, comprimidos y expandidos.
- Velocidad: 1ØØ caracteres por segundo.
- Comunicación por Interface RS232.

La nueva impresora también es apta para el Spectrum 128K sin ningún tipo de complemento (recordemos que éste lleva incluido un Interface RS232).

Otros complementos dignos de reseñar para el QL son:

- Unidad de floppys para discos de 3.5 pulgadas con capacidad de 1 Megabyte (72Ø K formateados).
- Monitores en color (BM 1Ø1Ø E y BM 4Ø4Ø P) y monocromático (BM 12 G) de marca BMC/INVESTRONICA.

SOFTWARE: POCAS NOVEDADES

A nivel de Software la verdad es que no ha habido demasiadas novedades, aunque algunas compañías se empeñen en hacernos creer lo contrario.

Ventamatic estaba allí, como otras veces (al fin y al cabo son vecinos del lugar), con sus productos de siempre y también con lo que ellos denominan la presentación oficial de su Logo en Castellano y la Tortuga- Robot que ya llevan algunos meses en el mercado.

Serma presentaba su nuevo catálogo para MSX, tras la reciente adquisición de los derechos de distribución de Konami para este ordenador, del cual lo más destacable es el Tenis. Para Spectrum, las novedades destacables eran el programa «KARATE» y la colección «Aprende a leer en Inglés».



Una de las nuevas compañías de software, recientemente constituida, MIND GAMES ESPAÑA, estaba también en SONIMAG con sus nuevos productos, si bien alguno de ellos viene ya de tiempo atrás, como el caso de «ALIEN» o «FANTASTIC VOYAGE». La compañía presentó, además, alrededor de unos 20 títulos contando los de la empresa de software «CENTURY CITY SOFTWARE».

ERBE anunciaba a bombo y platillo sus dos nuevos productos, «RAMBO» basados en la película de Silvester Stallone y el Basketball de Imagine, uno de los juegos más esperados y que está dentro de la nueva serie de la compañía «WORLD SERIES».

En definitiva, puede decirse que SONIMAG 85 ha estado pobre en novedades (sólo el 10% de la muestra estaba dedicada a ordenadores) y que estas novedades no lo han sido tanto después de haber asistido a la P. C. Show de Londres.

Lo mismo pero a escala nacional.



ANARANJADO

Como todos sabéis, el Specrum tiene una gama de colores limitada a los puros.

Pues bien, si queremos conseguir un color «intermedio» como es el anaranjado, podréis hacerlo mediante este truco de Fco. José Navas.

```
  10 FOR n=0 TO 7 STEP 2
  15 FOR m=1 TO 6 STEP 2
  20 POKE USR "c"+n,BIN 10101010
  25 POKE USR "c"+m,BIN 01010101
  30 NEXT m: NEXT n
  35 BORDER 7: PAPER 2: INK 6: C
LS
  40 PRINT " ": STOP
```

SIMULAR EL COPY

Eduardo Orts nos envía un truco para simular el COPY de la impresora, con una salvedad. Como en muchas ocasiones no nos interesa copiar toda la pantalla, sino sólo una zona, podemos delimitarla variando las variables f (filas) y c (columnas).

```
   5 REM COPY de baja resolucion
  10 FOR f=0 TO 21
  20 FOR c=0 TO 31
  30 LET a$=SCREEN$ (f,c)
  40 LPRINT a$;
  50 NEXT c
  60 LPRINT CHR$ 27+CHR$ 60: REM
Retorno del carro
  70 NEXT f
```

PRODUCIR EFECTOS

Podemos usar estos RANDOMIZES para producir bonitos efectos.

RANDOMIZE USR 5050
RANDOMIZE USR 4710

Pero si queremos conseguir efectos impresionantes, lo podremos hacer con este truco.

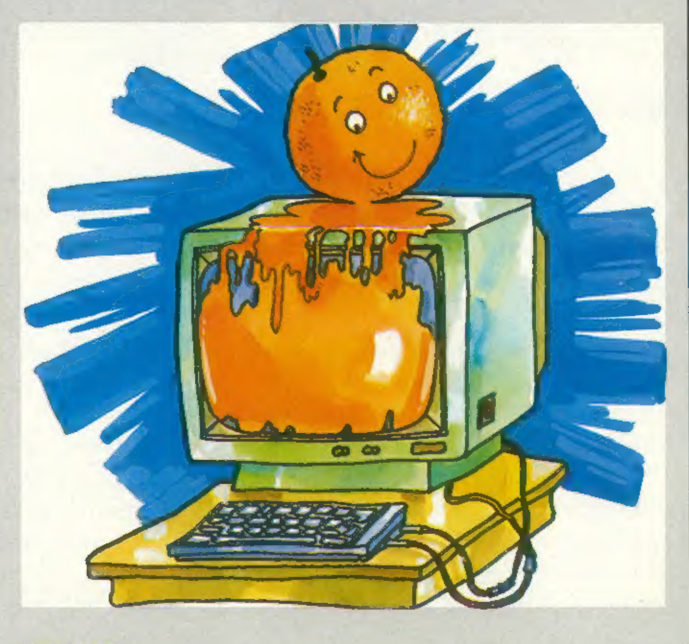
Nos lo manda José García Córdoba.

Para producir otro efecto

```
  10 PLOT 140,20: DRAW OVER 1;55
,55,51↑4: PLOT 130,65: DRAW 20,2
0,51↑4: STOP
```

«de relieve» Fernando Arderius nos ha enviado este otro truco que se puede mejorar con las funciones OVER, INVERSE.

```
   1 INK 3
  10 FOR x=7 TO 127
  20 PLOT x,70+70+SIN (.050*PI+x
)
  30 DRAW x,-100
  40 PLOT (255-x),70+70+SIN (.05
0*PI+x)
  50 DRAW -x,-100
  60 NEXT x
```



CRONOMETRO

No, no nos hemos equivocado, ni tampoco nos repetimos, porque, si bien en el número pasado hablamos de cómo hacernos un cronómetro con nuestro Spectrum, con este truco que nos manda Bernardo Tena, conseguiremos el mismo efecto, pero de otra manera.

```
  10 REM CRONOMETRO
  20 PRINT AT 5,0;"A PARA PARAR"
;AT 6,0;"S PARA EMPEZAR";AT 7,0;
"D PARA SEGUIR"
  30 PLOT 0,175: DRAW 111,0: DRA
W 0,-31: DRAW -111,0: DRAW 0,31
  40 PRINT AT 2,0;"s/10";AT 2,5;
"seg.";AT 2,10;"min."
  50 LET v=0
  60 LET m=0
 100 FOR n=0 TO 9: PRINT AT 1,1;
n: PAUSE 2.7
 150 IF n=9 THEN LET v=v+1: PRIN
T AT 1,5;v
 160 IF v=60 THEN PRINT AT 1,5;"
0": LET v=0: LET m=m+1: PRINT AT
 1,0;m
 200 IF INKEY$="a" OR INKEY$="A"
 THEN PAUSE 0
 300 IF INKEY$="s" OR INKEY$="S"
 THEN RUN
 400 NEXT n
 500 GO TO 100
```

SALVAR LAS TECLAS

El problema del deterioro de las teclas de nuestro ordenador es un hecho que motiva más de una consulta en nuestra sección de consultorio. Pues bien, para Javier de la Guardia solucionarlo ha sido de lo más fácil: coger esmalte de uñas incoloro (o transparente) y poner un poco en las teclas con lo que se formará una película protectora que evitará su borrado.

En este espacio también tienen cabida los trucos que nuestros lectores quieran proponer.
Para ello, no tienen más que enviarlos por correo a MICROHOBBY, C/ La Granja, 8. Polígono Industrial de Alcobendas (Madrid).

LA MOLE

Fernando de la PUENTE

Spectrum 48 K

NOTAS GRAFICAS

A B C D E F G H I J K L M N O P Q R

No es un programa de gigantes. Tampoco de cabezudos. Es simplemente, un entretenido juego laberíntico y aventurero que nos convertirá en buscadores de tesoros.

El objetivo es recoger los diversos componentes de un tesoro (vasijas, joyas y demás preciosidades) que se encuentran en una cámara laberíntica.

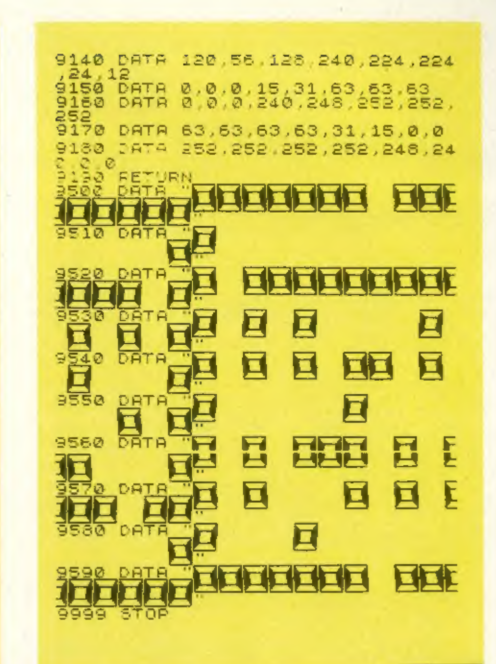
Pero la aventura no acaba aquí (sería demasiado fácil) ya que nuestro empeño por conseguir botín se verá ampliamente dificultado por la presencia y custodia de un gigantesco ser terriblemente peligroso con el que podemos toparnos en cualquier esquina: La Mole.

Si evitamos su presencia y damos con el pasaje secreto que nos conducirá rápidamente a la salida, podremos darnos por satisfechos, lo habremos conseguido.

```
   5 GO SUB 9000
  10 OVER 0: BORDER 0: PAPER 0:
INK 9: CLS
  15 GO SUB 7000
  20 DIM a$(10,64)
  24 LET sc=0: LET li=3: LET ite
ms=3
  25 LET t=0: LET x=17: LET y=2:
 LET ax=3: LET ay=28: LET items=
items+1
  30 RESTORE 9500: FOR a=1 TO 10
  READ a$(a): NEXT a
  40 GO SUB 600
  41 GO SUB 500
  42 GO SUB 550
 110 IF INKEY$="p" THEN GO SUB 5
00: LET y=y+(2 AND a$(((x+1)/2),
y+3)<>"▛"): BEEP .001,y+20: GO S
UB 500
 115 IF INKEY$="o" THEN GO SUB 5
00: LET y=y-(2 AND a$((x+1)/2,y)
<>"▜"): BEEP .001,50-y: GO SUB 5
00
 120 IF INKEY$="a" THEN GO SUB 5
00: LET x=x+(2 AND a$((x/2)+1,y+
1)<>"▛"): BEEP .001,40-y: GO SUB
 500
 125 IF INKEY$="q" THEN GO SUB 5
00: LET x=x-(2 AND a$(((x+1)/2)-
1,y+1)<>"▛"): BEEP .001,45-y: GO
 SUB 500
 130 IF a$(((x+1)/2),y+1)="" TH
EN LET t=t+1: LET sc=sc+100: FOR
 z=10 TO 40 STEP 5: BEEP .005,z:
 NEXT z: LET a$((x+1)/2,y+1)=" "
: LET a$(((x+1)/2)+1,y+33)=" ":
PRINT OVER 1;AT x,y;"";AT x+1,y
;""
 135 IF a$(((x+1)/2),y+1)="" TH
EN LET t=t+1: LET sc=sc+200: FOR
 z=20 TO 60 STEP 5: BEEP .005,z:
 NEXT z: LET a$((x+1)/2,y+1)="
": LET a$(((x+1)/2)+1,y+33)=" "
: PRINT OVER 1;AT x,y;"";AT x+
1,y;""
 140 IF t>=items AND (x<=2 OR x>
=19) AND y=14 THEN GO TO 700
 145 IF x>=19 AND y=14 THEN GO S
UB 500: LET x=3: LET y=14: GO SU
B 500
 150 IF x<=2 AND y=14 THEN GO SU
B 500: LET x=17: LET y=14: GO SU
B 500
 155 PRINT AT 21,7; PAPER 2; BRI
GHT 1;sc; BRIGHT 0;" "; BRIGHT 1
;AT 21,27;li; BRIGHT 0;" "
 160 IF x=ax AND y=ay THEN GO SU
B 800
 161 LET o=ax: LET p=ay
 165 IF ay>y THEN GO SUB 550: LE
T ay=ay-(2 AND a$((ax+1)/2,ay)<>
"▜"): GO SUB 550
 170 IF ay<y THEN GO SUB 550: LE
T ay=ay+(2 AND a$((ax+1)/2,ay+3)
<>"▛"): GO SUB 550
 175 IF ax<19 THEN IF ax>x THEN
GO SUB 550: LET ax=ax-(2 AND a$(
((ax+1)/2)-1,ay+1)<>"▛"): GO SUB
 550
 180 IF ax>2 THEN IF ax<x THEN G
O SUB 550: LET ax=ax+(2 AND a$((
ax+1)/2+1)<>"▛"): GO SUB 550
 190 IF ax=o AND ay=p THEN GO SU
B 550: LET ay=ay+(2 AND a$(((ax+
1)/2),ay+3)<>"▛")-(2 AND a$(((ax
+1)/2),ay)<>"▜"): GO SUB 550
 200 GO TO 110
 500 PRINT AT x,y; OVER 1; BRIGH
T 1; INK 6;""
 505 PRINT AT x+1,y; OVER 1; BRI
GHT 1; INK 6;""
 510 RETURN
 550 PRINT AT ax,ay; OVER 1; BRI
GHT 1; INK 4;""
 555 PRINT AT ax+1,ay; OVER 1; B
RIGHT 1; INK 4;""
 560 RETURN
 600 FOR a=1 TO 10: BEEP .001,40
+a: PRINT AT (a*2)-1,0; INK 4;;a
$(a): NEXT a
 601 PRINT AT 21,0; PAPER 2;"SCO
RE  ";sc;AT 21,21;"VIDAS  ";li
 605 LET z$="  ": FOR z=1 TO i
tems
 610 LET rx=(INT (RND*8)+1)*2: L
ET ry=(INT (RND*14)+1)+2
 615 IF a$((rx+2)/2,ry+1)<>" " T
HEN GO TO 610
 620 PRINT INK INT (RND*6)+2;AT
rx+1,ry;z$( TO 2);AT rx+2,ry;z$(
3 TO )
 621 LET a$((rx+2)/2,ry+1)=z$(1)
: LET a$((rx+2)/2,ry+2)=z$(2)
 622 LET a$((rx+1)/2,ry+32)=z$(3
): LET a$((rx+1)/2,ry+32)=z$(4)
 625 LET z$=("" AND z>=2)+("
  " AND z<=1): NEXT z
 630 RETURN
```



```
 700 FOR a=1 TO 50
 701 OUT 254,7
 702 OUT 254,130
 703 OUT 254,30
 704 OUT 254,16
 705 OUT 254,255
 706 OUT 254,201
 707 OUT 254,150
 708 NEXT a
 710 BEEP .002,a
 715 FOR a=0 TO 10: FOR b=1 TO 7
: BEEP .02,(a+b)+RND*10: PRINT A
T 10,11; PAPER b; INK 9; FLASH 1
;"BONUS": NEXT b: NEXT a
 720 CLS : LET sc=sc+500
 730 GO TO 25
 800 FOR b=1 TO 5: FOR a=0 TO 7:
 BORDER a: BEEP .005,a*b: NEXT a
: NEXT b: BORDER 0
 801 CLS
 805 LET li=li-1
 810 PRINT AT 10,8; FLASH 1; PAP
ER 2;"TE QUEDAN ";li;" VIDAS"
 815 FOR a=1 TO 100: PAUSE 1: NE
XT a
 816 IF li<=0 THEN GO TO 830
 820 CLS : GO TO 25
 830 CLS : PRINT INK 9;AT 10,9;
PAPER 3; FLASH 1;" JUEGO TERMINA
DO "
 840 PRINT AT 21,0;" PULSA UNA T
ECLA PARA JUGAR ": PAUSE 0: RUN
7000 LET b$="  ****EN BUSCA DEL
TESORO****  "
7010 FOR a=2 TO 31: FOR b=1 TO 2
1: PRINT AT b,a-1;b$( TO 33-a):
NEXT b: IF a<2 THEN NEXT a
7015 PRINT AT 21,0;"PULSA UNA TE
CLA"
7020 PAUSE 0: CLS
8000 PRINT AT 3,0;
8010 PRINT PAPER 6; INK 1;"  El
objetivo del juego es reco-ger t
odos los objetos que se en-cuent
ran esparcidos por el labe-rinto
.Para ello deberemos esca- par d
e la MOLE que protege el - tesor
o del laberinto.Posees tres vida
s que iras perdiendo si ca-es en
 manos de la MOLE.          Hay
 un pasage secreto que pue-des a
provechar para escapar de -la MO
LE y en el que puedes pa-  sar a
l siguiente nivel tras ha- ber r
ecogido todos los objetos. "
8015 PRINT "TECLAS:      /O/ IZQUIE
RDA                      /P/ DERECH
A                        /Q/ SUBIR
                         /A/ BAJAR
          "
8019 PRINT '"PULSA UNA TECLA"
8020 PAUSE 0: CLS
9000 RESTORE 9010: FOR a=USR "a"
 TO USR "r"+7: READ c: POKE a,c:
 NEXT a
9010 DATA 255,128,192,224,255,24
0,240,240
9020 DATA 254,2,6,14,254,30,30,3
0
9030 DATA 240,240,240,255,224,19
2,128,255
9040 DATA 30,30,30,254,14,6,2,25
4
9050 DATA 0,0,0,8,8,28,28,28
9060 DATA 28,28,62,29,29,30,0,0
9070 DATA 0,0,0,15,16,32,64,63
9080 DATA 0,0,0,240,16,8,4,252
9090 DATA 64,64,32,16,8,4,2,1
9100 DATA 2,4,8,16,32,64,128,0
9110 DATA 63,63,32,255,127,99,99
,103
9120 DATA 248,248,8,254,248,16,2
52,252
9130 DATA 79,79,64,63,7,31,24,16
```



```
9140 DATA 120,56,128,240,224,224
,24,12
9150 DATA 0,0,0,15,31,63,63,63
9160 DATA 0,0,0,240,248,252,252,
252
9170 DATA 63,63,63,63,31,15,0,0
9180 DATA 252,252,252,252,248,24
0,0,0
9190 RETURN
9500 DATA "
"
9510 DATA "
"
9520 DATA "
"
9530 DATA "
"
9540 DATA "
"
9550 DATA "
"
9560 DATA "
"
9570 DATA "
"
9580 DATA "
"
9590 DATA "
"
9999 STOP
```

COMBINATORIA

Pedro SILOS

Spectrum 16 K

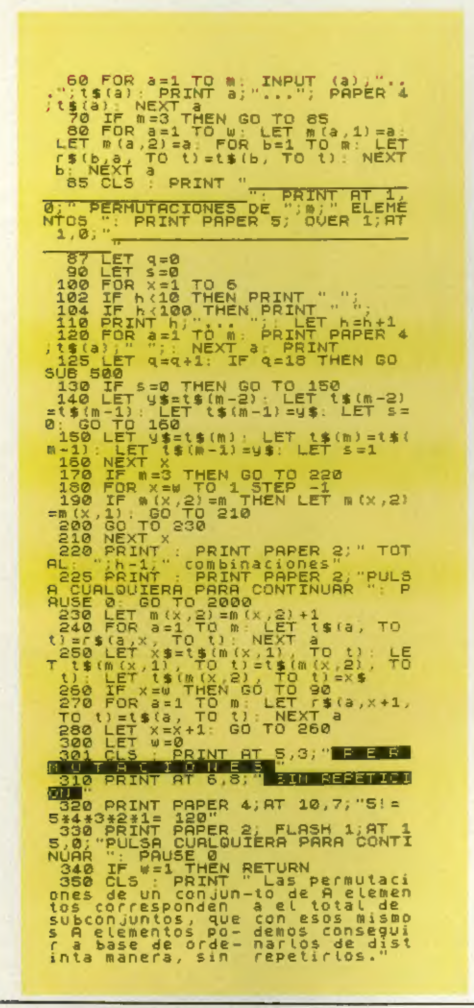
Con este programa podrás realizar todo tipo de combinaciones posibles con la utilización de sus mismos elementos ordenados de tal manera que no lleguen a repetirse.

Es decir, si tenemos un conjunto con cinco elementos, podemos ordenarlos de 120 maneras diferentes, información que redibirás con este programa con tan sólo contestar a sus preguntas.

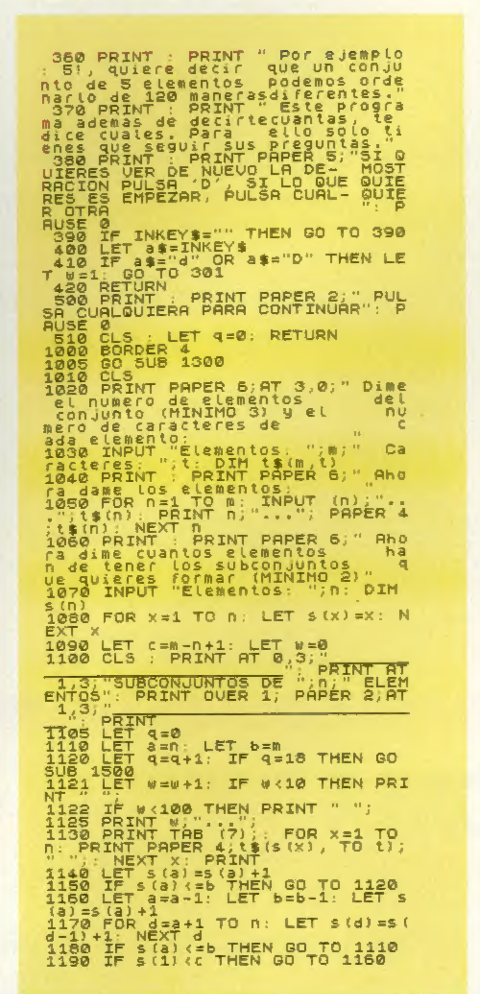
Las combinaciones de un número (A) sobre otro (B) corresponde al total de subconjuntos de B elementos que podemos obtener del total de A elementos sin repetir ninguno. Nada más fácil.



```
   4 REM ** © Pedro Silos Die **
   6 GO TO 2000
  10 GO SUB 300
  15 BORDER 4: CLS
  20 PRINT PAPER 6;AT 3,0;"  Dim
e el numero de elementos    (CO
MO MINIMO 3) y el numero de   ca
racteres de cada uno de        e
llos:                          "
  30 INPUT "Elementos: ";m;" Car
acteres: ";t: LET h=1: LET w=m-3
  40 DIM t$(m,t)
  45 IF m<>3 THEN DIM m(w,2): DI
M r$(m,w,t)
  50 PRINT : PRINT PAPER 6;" Aho
ra dame los elementos:         "
```



```
  60 FOR a=1 TO m: INPUT (a);"..
.";t$(a): PRINT a;"..."; PAPER 4
;t$(a): NEXT a
  70 IF m=3 THEN GO TO 85
  80 FOR a=1 TO w: LET m(a,1)=a:
 LET m(a,2)=a: FOR b=1 TO m: LET
 r$(b,a, TO t)=t$(b, TO t): NEXT
 b: NEXT a
  85 CLS : PRINT "________________
________________": PRINT AT 1,
0;" PERMUTACIONES DE ";m;" ELEME
NTOS ": PRINT PAPER 5; OVER 1;AT
 1,0;"________________________
_______"
  87 LET q=0
  90 LET s=0
 100 FOR x=1 TO 6
 102 IF h<10 THEN PRINT " ";
 104 IF h<100 THEN PRINT " ";
 110 PRINT h;"... ";: LET h=h+1
 120 FOR a=1 TO m: PRINT PAPER 4
;t$(a);" ";: NEXT a: PRINT
 125 LET q=q+1: IF q=18 THEN GO
SUB 500
 130 IF s=0 THEN GO TO 150
 140 LET y$=t$(m-2): LET t$(m-2)
=t$(m-1): LET t$(m-1)=y$: LET s=
0: GO TO 160
 150 LET y$=t$(m): LET t$(m)=t$(
m-1): LET t$(m-1)=y$: LET s=1
 160 NEXT x
 170 IF m=3 THEN GO TO 220
 180 FOR x=w TO 1 STEP -1
 190 IF m(x,2)=m THEN LET m(x,2)
=m(x,1): GO TO 210
 200 GO TO 230
 210 NEXT x
 220 PRINT : PRINT PAPER 2;" TOT
AL: ";h-1;" combinaciones"
 225 PRINT : PRINT PAPER 2;"PULS
A CUALQUIERA PARA CONTINUAR ": P
AUSE 0: GO TO 2000
 230 LET m(x,2)=m(x,2)+1
 240 FOR a=1 TO m: LET t$(a, TO
t)=r$(a,x, TO t): NEXT a
 250 LET x$=t$(m(x,1), TO t): LE
T t$(m(x,1), TO t)=t$(m(x,2), TO
 t): LET t$(m(x,2), TO t)=x$
 260 IF x=w THEN GO TO 90
 270 FOR a=1 TO m: LET r$(a,x+1,
TO t)=t$(a, TO t): NEXT a
 280 LET x=x+1: GO TO 260
 300 LET w=0
 301 CLS : PRINT AT 5,3;" P E R
M U T A C I O N E S "
 310 PRINT AT 6,8;" SIN REPETICI
ON "
 320 PRINT PAPER 4;AT 10,7;"5!=
5*4*3*2*1= 120"
 330 PRINT PAPER 2; FLASH 1;AT 1
5,0;"PULSA CUALQUIERA PARA CONTI
NUAR ": PAUSE 0
 340 IF w=1 THEN RETURN
 350 CLS : PRINT " Las permutaci
ones de un conjun-to de A elemen
tos corresponden  a el total de
subconjuntos, que con esos mismo
s A elementos po- demos consegui
r a base de orde- narlos de dist
inta manera, sin  repetirlos."
```



```
 360 PRINT : PRINT " Por ejemplo
: 5!, quiere decir  que un conju
nto de 5 elementos  podemos orde
narlo de 120 manerasdiferentes."
 370 PRINT : PRINT " Este progra
ma ademas de decirtecuantas, te
dice cuales. Para   ello solo ti
enes que seguir sus preguntas."
 380 PRINT : PRINT PAPER 5;"SI Q
UIERES VER DE NUEVO LA DE-  MOST
RACION PULSA 'D', SI LO QUE QUIE
RES ES EMPEZAR, PULSA CUAL- QUIE
R OTRA                        ": P
AUSE 0
 390 IF INKEY$="" THEN GO TO 390
 400 LET a$=INKEY$
 410 IF a$="d" OR a$="D" THEN LE
T w=1: GO TO 301
 420 RETURN
 500 PRINT : PRINT PAPER 2;" PUL
SA CUALQUIERA PARA CONTINUAR": P
AUSE 0
 510 CLS : LET q=0: RETURN
1000 BORDER 4
1005 GO SUB 1300
1010 CLS
1020 PRINT PAPER 6;AT 3,0;" Dime
 el numero de elementos      del
 conjunto (MINIMO 3) y el     nu
mero de caracteres de          c
ada elemento:                  "
1030 INPUT "Elementos: ";m;"  Ca
racteres: ";t: DIM t$(m,t)
1040 PRINT : PRINT PAPER 6;" Aho
ra dame los elementos:         "
1050 FOR n=1 TO m: INPUT (n);"..
.";t$(n): PRINT n;"..."; PAPER 4
;t$(n): NEXT n
1060 PRINT : PRINT PAPER 6;" Aho
ra dime cuantos elementos     ha
n de tener los subconjuntos    q
ue quieres formar (MINIMO 2)"
1070 INPUT "Elementos: ";n: DIM
s(n)
1080 FOR x=1 TO n: LET s(x)=x: N
EXT x
1090 LET c=m-n+1: LET w=0
1100 CLS : PRINT AT 0,3;"________
_____________": PRINT AT
 1,3;"SUBCONJUNTOS DE ";n;" ELEM
ENTOS": PRINT OVER 1; PAPER 2;AT
 1,3;"_________________________
 ": PRINT
1105 LET q=0
1110 LET a=n: LET b=m
1120 LET q=q+1: IF q=18 THEN GO
SUB 1500
1121 LET w=w+1: IF w<10 THEN PRI
NT " ";
1122 IF w<100 THEN PRINT " ";
1125 PRINT w;"...";
1130 PRINT TAB (7);: FOR x=1 TO
n: PRINT PAPER 4;t$(s(x), TO t);
" ";: NEXT x: PRINT
1140 LET s(a)=s(a)+1
1150 IF s(a)<=b THEN GO TO 1120
1160 LET a=a-1: LET b=b-1: LET s
(a)=s(a)+1
1170 FOR d=a+1 TO n: LET s(d)=s(
d-1)+1: NEXT d
1180 IF s(a)<=b THEN GO TO 1110
1190 IF s(1)<c THEN GO TO 1160
```

```
1200 PRINT : PRINT PAPER 2;" TOT
AL: ";w;" subconjuntos"
1205 PRINT : PRINT PAPER 2;"PULS
A CUALQUIERA PARA CONTINUAR ": P
AUSE 0: GO TO 2000
1300 LET w=0
1301 CLS
1310 PRINT AT 4,3;" C O M B I N
A C I O N E S "
1315 PRINT AT 5,8;" SIN REPETICI
ON "
1320 PRINT PAPER 4;AT 10,2;"/5\
  5!    5*4*3*2*1  20 "
1330 PRINT PAPER 4; OVER 1;AT 10
,2;"   = _____= __________= __= 1
0"
1340 PRINT PAPER 4;AT 11,2;"\3/
 2!*3!  2*1*3*2*1    2      "
1350 PRINT PAPER 2; FLASH 1;AT 1
5,2;" PULSA CUALQIERA PARA SEGUI
R ": PAUSE 0
1355 IF w=1 THEN RETURN
1360 CLS : PRINT " Como sabras,
las COMBINACIONES de un numero (
A) sobre otro (B),corresponden a
l total de subcon_juntos de B el
ementos que pode- mos obtener de
l total de A ele_ mentos, SIN RE
PETIR NINGUNO."
1370 PRINT " Por ejemplo: /5\"
1380 PRINT "              \3/  q
uiere decir en realidad cuantos
subconjuntosde 3 elementos podem
os realizar con un conjunto de 5
 elementos, que como habras vist
o en la de- mostracion son 10."
1390 PRINT " Este programa adema
s de decirtecuantos, te dice cua
les, para   ello solo tienes que
 seguir sus preguntas."
1400 PRINT PAPER 5;" SI QUIERES
VER DE NUEVO LA DE- MOSTRACION P
ULSA 'D', SI LO QUE QUIERES ES E
MPEZAR PULSA CUAL-  QUIER OTRA
                    "
1410 IF INKEY$="" THEN GO TO 141
0
1415 LET a$=INKEY$
1420 IF a$="d" OR a$="D" THEN LE
T w=1: GO TO 1301
1430 RETURN
1500 PRINT PAPER 2;" PULSA CUALQ
UIERA PARA CONTINUAR"
1510 PAUSE 0: CLS : LET q=0: RET
URN
2000 CLEAR : CLS
2010 PRINT PAPER 4;AT 5,3;" COMB
INACIONES.........1"
2020 PRINT PAPER 4;AT 8,3;" PERM
UTACIONES.........2"
2025 PRINT PAPER 3;AT 11,3;" FIN
 DEL PROGRAMA.......3"
2030 PRINT PAPER 2;AT 15,0;"PULS
A LA TECLA CORRESPONDIENTE
 A LO QUE QUIERAS HACER       "
2040 IF INKEY$="" THEN GO TO 204
0
2050 LET a$=INKEY$
2060 IF a$="1" THEN GO TU 1000
2070 IF a$="2" THEN GO TJ 1
2075 IF a$="3" THEN STOF
2080 GO TO 2040
```

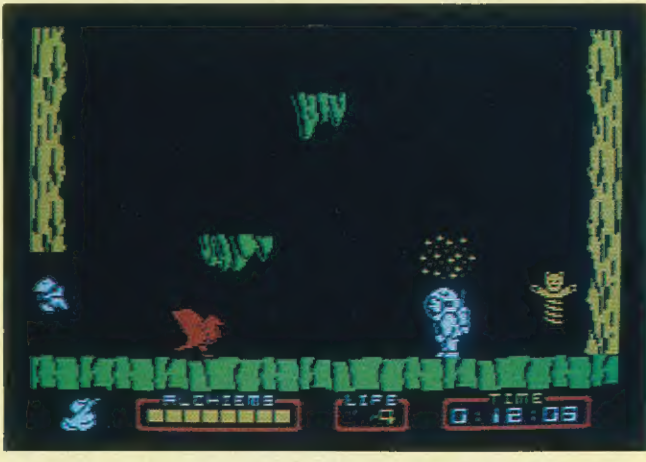
¡NUEVO!

Paseo Lunar

NODES OF YESOD

Odín
Serma
Videoaventura
P.V.P.: 2.800



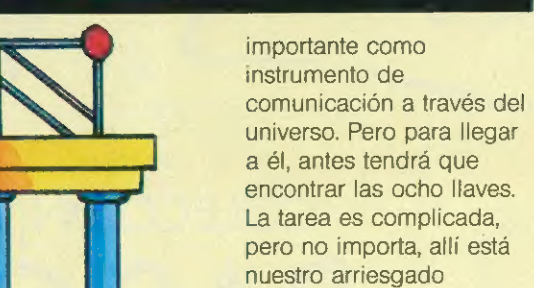


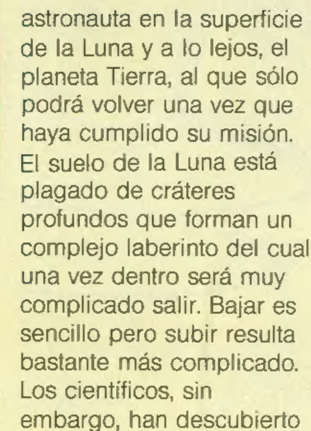
Nodes of Yesod hacía su aparición publicitaria hace ya algún tiempo, con una campaña muy propia del estilo de Ultimate. Sin pistas, sin frases de ningún tipo, con un velado misterioso que lo envolvía todo y que parecía estar aguardando el momento en el que el jugador descubriera lo que se escondía en realidad detrás de una bonita presentación.

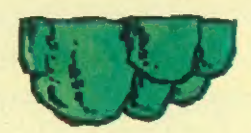
Y el momento llegó. Allí estaba nuestro personaje, un astronauta llamado Charlie que había llegado a la superficie de aquel planeta con el fin de descubrir el misterio que se encontraba oculto en las entrañas de éste.

Charlie es enviado desde la Tierra a la Luna para encontrar una especie de monolito que, según los científicos, es muy importante como instrumento de comunicación a través del universo. Pero para llegar a él, antes tendrá que encontrar las ocho llaves. La tarea es complicada, pero no importa, allí está nuestro arriesgado astronauta en la superficie de la Luna y a lo lejos, el planeta Tierra, al que sólo podrá volver una vez que haya cumplido su misión.









El suelo de la Luna está plagado de cráteres profundos que forman un complejo laberinto del cual una vez dentro será muy complicado salir. Bajar es sencillo pero subir resulta bastante más complicado. Los científicos, sin embargo, han descubierto la existencia de unos seres que habitan el planeta y que, van a ser muy importantes en la misión de Charlie. Se trata de una especie de topos capaces de devorar los muros de las cavernas cuando éstos escondan tras de sí otros lugares. Lo primero que hay que hacer es, por tanto, encontrar alguno de estos bichos y utilizarlo posteriormente cuando nos sean necesarios.















Y de este modo, comienza la aventura. Charlie recorre la luna atento, esperando encontrar en alguno de esos cráteres a uno de los topos. Hay que estar muy vigilante, puede aparecer en cualquier momento y disponemos de muy poco tiempo para hacernos con él. Si pulsamos el botón de salto veremos enseguida cómo nuestro personaje desafía las leyes de la gravedad y avanza por el aire en una simpática pirueta. Ese salto puede ayudarnos en un momento determinado a capturar al topo y es, además, nuestra única arma para ir de un lado a otro, una vez en el interior de las cavernas.

Cuando ya tengamos al topo en nuestro poder estaremos en condiciones de empezar a examinar el interior de los cráteres, pero mucho cuidado, un salto mal calculado podría adelantar nuestra incursión y un mal paso podría enviarnos a la parte más profunda de la enorme caverna. Aunque el traje de astronauta está preparado para resistir muchos golpes, una caída desde una altura excesiva acabaría con una de las vidas de Charlie.

Los enemigos de las profundidades son muchos y habrá algunos a los que podremos derrotar y otros con los que hay que tener mucho cuidado. También hay una serie de zonas que son una especie de transbordadores que nos llevarán a otro lugar dentro de la caverna.

Después de algún tiempo de juego nos damos cuenta de que aún queda mucho camino por recorrer y que hay que intentarlo de la forma más segura posible.



RESUMIENDO...

Originalidad: Nos recuerda mucho a otros tiempos de Ultimate, sin embargo, hay detalles en el programa llenos de originalidad que no se pueden negar, como es el caso del «topo».
* * *

Gráficos: Muy buenos y aunque el interior de las cavernas nos recuerda a Underwulde hay que reconocer el mérito en el diseño de la pantalla y, sobre todo, lo brillante que resulta el movimiento del astronauta.
* * * *

Sonido: Los efectos de rigor y una melodía al principio del juego llena de ritmo.
* * *

Valoración: De esos juegos de los de pasarlo bien a los que es muy difícil renunciar antes de haber llegado al final. El nivel de adicción es bastante alto.
* * * *

Operación Market Garden

ARNHEM

CCS
Wargame
P.V.P.: 2.900

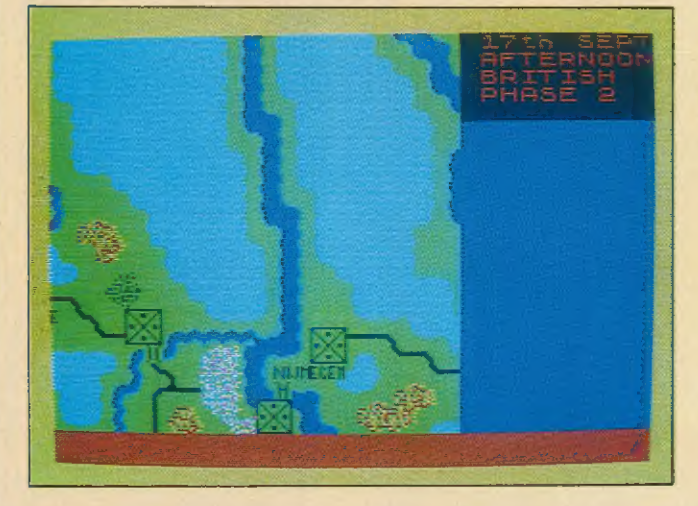


Los Wargames son un género de juegos muy difundido en el mundo, aunque quizás en nuestro país aún estemos empezando a descubrirlo a nivel popular. Los ingleses, que son muy aficionados a los juegos de estrategia, han convertido su afición favorita en programas para ordenador, aunque la verdad, con escaso éxito fuera de sus fronteras y a veces incluso dentro de ellas.

ARNHEM es la reproducción de la histórica operación Market Garden llevada a cabo en 1944 según los planes del general Montgomery con el fin de facilitar la ocupación de Holanda por parte del ejército aliado.

Cuando empieza el juego accedemos a un menú donde se nos da a elegir entre cinco opciones diferentes, para comenzar en el punto de la misión que nosotros prefiramos. Una vez que nos hemos decidido, tendremos que mover nuestras tropas hacia el objetivo previsto. El programa viene acompañado de un plano general que nos sirve para situarnos en el lugar donde se desarrolla la acción. Todos los comandos y movimientos aparecen durante todo el juego en la parte izquierda de la imagen, con el fin de que nos sea mucho más sencillo su utilización.

Hay cinco escenarios distintos.



RESUMIENDO...

Originalidad: Aporta un tratamiento mucho más simplificado que otros juegos parecidos, aunque la pantalla no difiere demasiado. * * *

Gráficos: Los normales en todo Wargame, podemos destacar el tratamiento del color y la claridad de las imágenes. * * *

Valoración: Un buen Wargame. * * *

Laberinto Peligroso

MONOLITH

Edge
Arcade
Importado



Detrás de una cuidada presentación, metido en una caja alargada en forma de monolito, se esconde un programa que nos sería muy difícil de situar en el tiempo. Tiene toda la apariencia de uno de aquellos juegos de laberintos de la primera época del Spectrum.

Cada una de las caras de la cinta tiene un programa distinto, pero lo cierto es que los dos son muy parecidos, tan sólo se ha modificado un poco el escenario.

Tanto el objetivo como la mecánica del juego son bastante simples. El programa genera al principio un laberinto y nuestra misión es la de dirigir por él a un curioso muñeco de ojos saltones y recoger unos rombos situados en algunas zonas de la pantalla. Visto así parece demasiado sencillo, el problema viene cuando descubrimos que al pasar por debajo de una especie de bolas, las amarillas, éstas se desprenden y pueden aplastar a nuestro personaje y ahí, precisamente ahí, es donde reside el meollo de la cuestión, ya que es muy difícil pasar esquivando a las contundentes piedras, y más aún porque hay que



hacerlo en un tiempo muy limitado que no nos permite pararnos a pensar demasiado.

Hay un marcador que nos indica cuánto nos queda para completar la misión sobre un total de 150 sg.

RESUMIENDO...

Originalidad: No lo es mucho, pero tiene algún detalle a considerar, que dota al programa de una personalidad propia. * *

Gráficos: Muy simples, sin alardes de ningún tipo y que recuerdan a otros juegos parecidos. * *

Sonido: Los normales sin demasiados excesos. *

Valoración: Un programa más propio de otra época. * *

Ninguno

CICLOS DE MEMORIA:

2

CICLOS DE RELOJ:

7

EJEMPLO:

```
LD A,(DE)
```

Contenido del registro acumulador, no significativo.

Contenido del par de registros DE

(D):	0 1 0 0 1 1 1 1	4Fh
(E):	1 1 1 1 1 1 1 1	FFh

Contenido de la posición de memoria 4FFFh

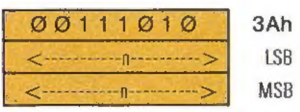
(4FFFh):	1 1 1 0 1 1 1 0	EEh

Ejecutamos la instrucción:

LD A,(DE): 0 0 0 1 1 0 1 0 1Ah

Contenido del acumulador después de la ejecución

(A):	1 1 1 0 1 1 1 0	77h



OBJETO:

Carga en el registro acumulador, el contenido de la posición de memoria direccionada por el operando "nn".

CODIGO MAQUINA:

0 0 1 1 1 0 1 0	3Ah
<----------n---------->	LSB
<----------n---------->	MSB

INDICADORES DE CONDICION:

Ninguno

CICLOS DE MEMORIA:

4

CICLOS DE RELOJ:

13

EJEMPLO:

```
CERO EQU #456A
LD A,(CERO)
```

La primera línea de este ejemplo define una etiqueta, esta operación no tiene código máquina y sirve simplemente, para indicarle al ensamblador, que allá donde le digamos la palabra "CERO", debe entender que queremos decir el número 456AH.

Esta instrucción también se podría haber escrito sin etiqueta de la siguiente forma:

```
LD A, (#456A)
```

La utilidad de las etiquetas es que si vamos a acceder a la posición 456Ah muchas veces, seguramente nos resulte más fácil recordar la palabra "CERO" que el número 456Ah.

Contenido de la posición de memoria 456Ah.

(456Ah):	0 0 0 0 0 0 0 0	00h

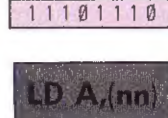
Ejecutamos la instrucción:

LD A, (#456A):	00111010	3Ah
	01101010	6Ah
	01000101	45h

Observe cómo se codifica el operando: el octeto de orden inferior (64h) se almacena, en la instrucción, delante del octeto de orden superior (45h).

Contenido del acumulador después de la ejecución

(A):	0 0 0 0 0 0 0 0	00h



OBJETO:

Carga en el acumulador, el contenido del registro "I" (vector de página de interrupción), y carga en el indicador "P/V" del registro "F", el estado del flip/flop de aceptación de interrupción "IFF2", que será "1" si la interrupción está habilitada y "Ø" si está inhibida. De esta forma, es posible comprobar de una sola instrucción, el estado del microprocesador en cuanto a las interrupciones.

CODIGO MAQUINA:

1 1 1 0 1 1 0 1	EDh
0 1 0 1 0 1 1 1	57h

INDICADORES DE CONDICION:

S (signo): Pone a "1" si "I" es negativo, es decir, si su bit de más peso es "1".

Z (cero): Pone a "1" si "I" vale cero.

H (semiacarreo): Pone a "0".

P/V (Paridad/rebosamiento): Pone a "1" si las interrupciones están habilitadas y a "0" si están inhibidas.

N (suma/resta): Pone a "0".

C (acarreo): Permanece con su estado anterior.

CICLOS DE MEMORIA:

2

Dónde ubicar un programa en C/M

en principio, un programa en código máquina se puede colocar en cualquier lugar de la memoria, de hecho, existen programas comerciales que la ocupan prácticamente por completo. No obstante, para nuestros fines existen zonas más adecuadas que otras.

Se supone que un programador aficionado, utilizará rutinas en C/M combinadas con un programa principal en Basic, por lo que habrá que respetar una zona de memoria para que el Basic pueda trabajar.

Básicamente, existen cuatro zonas donde situar nuestros programas:

1. Por encima de la RAMTOP.
2. En el buffer de impresora.
3. En el archivo de pantalla.
4. Dentro del programa Basic.

Veámosalas una por una:

1. **Por encima de la RAMTOP:** Es la zona más adecuada para colocar un programa en C/M, ya que queda protegido de borrados por el sistema Basic. En primer lugar, deberemos bajar la RAMTOP con el uso de CLEAR, como se veía en el ejemplo anterior. Una vez cargado nuestro programa, no podrá ser borrado ni siquiera con NEW; para volver a la situación inicial, deberemos teclear:

```
RANDOMIZE USR 0
```

Que si borrará el programa en C/M y todo lo que haya en la memoria del ordenador. Otra forma de destruir nuestro programa sería, volver a subir la RAMTOP.

2. **En el buffer de impresora:** Existe en la RAM, una zona reservada de 256 bytes, que empieza en la dirección 23296 (B00h) y acaba en la 23551 (5BFFh); esta zona la utiliza el Spectrum cuando trabaja con una impresora tipo ZX-Printer (Alphacom-32 o Seikosha GP-50S); si no va a utilizar ninguna de estas impresoras, puede almacenar en esta zona una rutina corta (256 bytes máximo) que no le ocupará, por tanto, memoria en la zona de programa. Tenga en cuenta, no obstante, que su rutina será borrada si utiliza los comandos: NEW, LPRINT, LLIST y COPY.

3. **En el archivo de pantalla:** En casos especiales, se utiliza el archivo de pantalla par almacenar programas en C/M, es una técnica usada en algunos copiadores para no ocupar memoria útil. Si no desea "ensuciar" la pantalla, puede poner los atributos correspondientes al mismo color de tinta y papel, con lo que los bytes no se visualizarán en forma de pixels. Cuando utilice esta técnica, tenga en cuenta que su programa puede ser corrompido por el uso de NEW, CLEAR y cualquier comando que afecte a la pantalla. El archivo de pantalla va desde 16384 (4000h) hasta 22527 (57FFh).

4. **Dentro del programa Basic:** Esta era la técnica usada en el ZX-81, consiste en hacer que la primera línea del programa sea una línea REM, con tantos espacios, como bytes tenga el programa C/M a almacenar. La dirección de inicio de esta zona es (PROG)+5. Este método tiene la ventaja de poder salvar juntos el Basic y el Código Máquina, si bien, su empleo no es recomendable si se tiene conectado el INTERFACE 1, ya que este dispositivo desplaza el programa Basic, y por tanto, nuestra rutina en C/M, a menos que ésta sea *reubicable* y entremos en ella, calculando cada vez la dirección de entrada a partir del contenido de la variable PROG. En este caso, nuestra rutina sólo se borra editando la línea, o borrando el programa Basic con NEW.

De todos éstos, el sistema usado con más frecuencia es el primero, y es el que usaremos en nuestros ejemplos. Si se tiene conectado un interface de impresora INDESCOMP, ha de tenerse en cuenta que su software ocupa los 1000 bytes más altos de la memoria.



LD dd,nn

El número nn de 2 bytes, es transferido al par de registros especificado por el operando dd.

Nnemónico: LD **Operandos:** dd,nn

Formato binario:

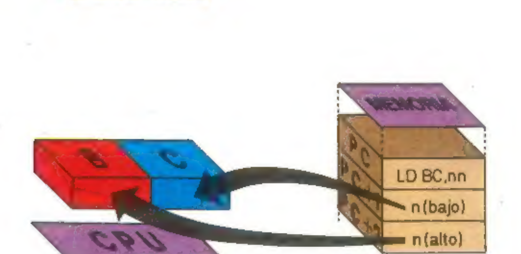
0 0 d d 0 0 0 1

n n n n n n n n

n n n n n n n n

Ciclos: 3
Estados: 10 (4,3,3)

Indicadores: ninguno



Instr.	Hex.	Dec.
LD BC,nn	01,n,n	1,n,n
LD DE,nn	11,n,n	17,n,n
LD HL,nn	21,n,n	33,n,n
LD SP,nn	31,n,n	49,n,n
LD IX,nn	DD,21,n,n	221,33,n,n
LD IY,nn	FD,21,n,n	253,33,n,n

Ejemplo:

Después de ejecutar la instrucción
LD BC,4000H
resultará que el par BC contiene 4000H.

El código del par dd, para la construcción del código binario de la instrucción es:

BC	00
DE	01
HL	10
SP	11

Giro Horizontal

Esta rutina realiza un giro de la pantalla tomando como eje una línea vertical situada en el centro de la misma.

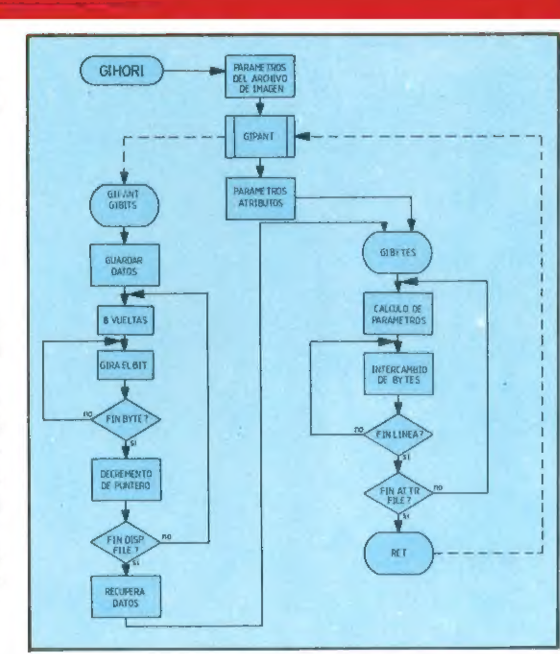
La forma de llamada es la usual, es decir:
RANDOMIZE USR n
siendo n la dirección a partir de la cual se ha situado la rutina (es relocatable).

Funcionamiento:

Utiliza la subrutina llamada GIPANT compuesta a su vez por otras dos subrutinas cuyos nombres son GIBITS y GBYTES.

La primera parte de la rutina trabaja en el fichero de pantalla, invirtiendo cada una de las 8 líneas de puntos de cada carácter sobre sí mismas, bit a bit (GIBITS), trasladándolas después a su dirección definitiva, al otro lado de la pantalla (GBYTES).

Por último intercambiará los atributos de los caracteres (CALL GBYTES), localizando su dirección en el fichero de atributos.



LD IX,nn

El número nn de 2 bytes, es transferido al par IX.

Mnemónico: LD **Operandos:** IX,nn

Formato binario:

Ciclos: 4
Estados: 14 (4,4,3,3)

Indicadores: ninguno

LD IY,nn

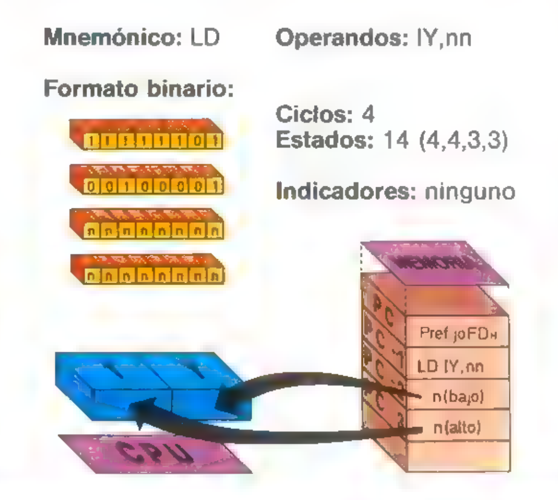
El número nn de 2 bytes, es transferido al par IY.

Mnemónico: LD **Operandos:** IY,nn

Formato binario:

Ciclos: 4
Estados: 14 (4,4,3,3)

Indicadores: ninguno



Un ensamblador es una herramienta de software (un programa), diseñado para simplificar las tareas que conlleva escribir los programas en código máquina, bien en binario o en hexadecimal.

El lenguaje ensamblador es una serie de nombres simbólicos (mnemónicos) de operación, fácilmente comprensibles, que se corresponden con las microinstrucciones de la CPU (Unidad Central de Proceso), lo cual obliga al programador de lenguaje ensamblador a conocer detalladamente cada una de las operaciones que ésta realiza.

Para usar el lenguaje ensamblador necesitamos un fichero de código fuente, que es una lista de líneas de texto, que deben cumplir las siguientes exigencias:

1. Número de línea, por cuyo orden son colocadas y ensambladas, a semejanza del Basic.

2. Campo de etiqueta, referencia necesaria para que el ensamblador desarrolle el flujo de programa deseado, en saltos u otras instrucciones que manejen direcciones.

3. Campo de código de operación (mnemónico), es opcional, y puede contener en lugar del código una directiva de ensamblador (pseudoinstrucción).

4. Campo de operando, también opcional, respetando la estructura del código mnemónico, puede tener ningún, uno o dos operandos, en este último caso deben ir separados por coma, y siempre que sean numéricos, pueden ser sustituidos por una expresión simbólica (con etiquetas).

5. Campo de comentario, opcional, de ayuda para entender mejor los programas, debe ir precedido de un punto y coma.

Todos los campos de una línea deben estar separados al menos por un espacio, siendo acon-

> Código fuente
> Código objeto
> Código máquina
> Líneas de ensamblador
> Campos
> Ensamblaje en 2 pasos

LD (IY+d),n LD (IY+d),r LD r,(IY+d)

```
10 ; *  GIRO HORIZONTAL  *
20          ORG    60000   ;RUTINA NO REUBICABLE
30 START    LD     HL,16384 ;Comienzo de pantalla
40          LD     DE,#1800 ,Long. DISPLAY FILE
50          CALL   GIPANT  ;Gira DISPLAY FILE
60          LD     DE,#300 ;Long. archivo atrib.
70          JR     GBYTES  ;Gira arch. atributos
80 GIPANT
90 GIBITS   PUSH   HL      ;Com. de pantalla
100         PUSH   DE      ;Long. arch. atribut.
110 BUCLE1  LD     B,8     ;No. de bits por byte
120         LD     A,(HL)
130 BUCLE2  RLA    ;       ,Extrae bit
140         RR     (HL)    ;Guarda bit
150         DJNZ   BUCLE2
160         INC    HL      ;Puntero
170         DEC    DE      ;Longitud
180         LD     A,E
190         OR     D
200         JR     NZ,BUCLE1;6144 vueltas
210         POP    DE      ;Recupera longitud
220         POP    HL      ;Recupera comienzo
230         LD     C,32    ;Anchura de linea
240 GBYTES  PUSH   HL      ;Puntero
250         PUSH   DE      ;Longitud
260         PUSH   BC      ;Anchura
270         LD     E,L
280         LD     D,H     ;Transfiere HL a DE
290         ADD    HL,BC   ;Incrementa anchura
300         DEC    ,HL     ;Puntero A
310         SRL    C       ;C/2
320 BUCLE3  LD     A,(HL)
330         LD     B,A     ;Cambia
340         LD     A,(DE)  ;contenido DE
350         LD     (HL),A  ;por cotenido de HL
360         LD     A,B
370         LD     (DE),A
380         DEC    HL      ;Puntero A
390         INC    DE      ;Puntero B
400         DEC    C       ;Ancho divid. entre 2
410         JR     NZ,BUCLE3
420         POP    BC      ;Ancho
430         POP    HL      ,Puntero
440         OR     A       ;Carry a 0
450         SBC    HL,BC   ;Resta ancho
460         EX     DE,HL   ;Lo transfiere a DE
470         POP    HL      ;Puntero
480         ADD    HL,BC   ;Suma ancho
490         LD     A,D
500         OR     E       ;Continua el bucle
510         JR     NZ,GBYTES;  si DE<>0
520         RET    ;       ;Si DE=0 fin
```

```
10 DATA "21 00 40 11 00 18 CD 6E",453
20 DATA "EA 11 00 03 18 14 E5 D5",740
30 DATA "06 08 7E 17 CB 1E 10 FB",663
40 DATA "23 1B 7B B2 20 F2 D1 E1",1071
50 DATA "0E 20 E5 D5 C5 5D 54 09",871
60 DATA "2B CB 39 7E 47 1A 77 78",765
70 DATA "12 2B 13 0D 20 F5 C1 E1",788
80 DATA "B7 ED 42 EB E1 09 7A B3",1256
90 DATA "20 E0 C9                ",457
```

LD (IY+d),n

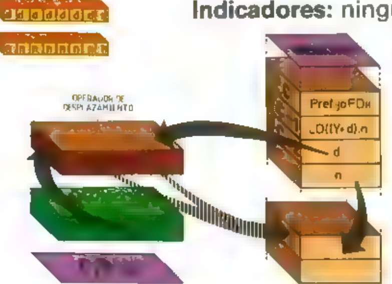
El número de 8 bits n es transferido a la dirección de memoria especificada por la suma del contenido del par IY y el desplazamiento d (d es un número de 8 bits en complemento a 2).

Mnemónico: LD **Operandos:** (IY+d),n

Formato binario:

Ciclos: 5
Estados: 19 (4,4,3,5,3)

Indicadores: ninguno

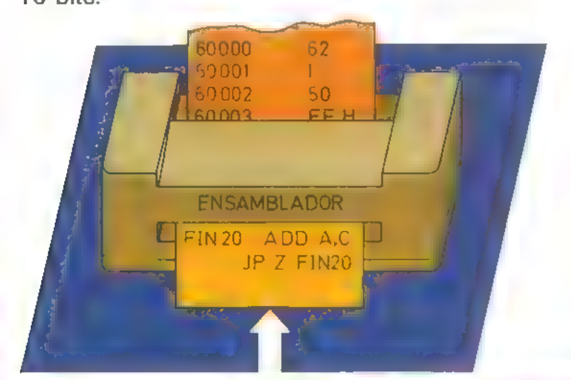




Instr.	Hex.	Dec.
LD (IY+d),n	FD,36,d,n	253,54,d,n
LD (IY+d),A	FD,77,d	253,119,d
LD (IY+d),B	FD,70,d	253,112,d
LD (IY+d),C	FD,71,d	253,113,d
LD (IY+d),D	FD,72,d	253,114,d
LD (IY+d),E	FD,73,d	253,115,d
LD (IY+d),H	FD,74,d	253,116,d
LD (IY+d),L	FD,75,d	253,117,d
LD A,(IY+d)	FD,7E,d	253,126,d
LD B,(IY+d)	FD,46,d	253,70,d
LD C,(IY+d)	FD,4E,d	253,78,d
LD D,(IY+d)	FD,56,d	253,86,d
LD E,(IY+d)	FD,5E,d	253,94,d
LD H,(IY+d)	FD,66,d	253,102,d
LD L,(IY+d)	FD,6E,d	253,110,d

sejable el empleo de tabulaciones, para que queden alineados por columnas, que contribuye al mejor entendimiento del programa.

- Una expresión numérica en lenguaje ensamblador es una combinación de números, símbolos y operadores, respetando las reglas algebraicas, donde cada elemento de la expresión es un término, y el resultado debe estar acorde con el operando a que sustituye, en su rango, de 8 a 16 bits.



Normalmente una expresión numérica debe poder admitir números en cualesquiera de las bases corrientemente utilizadas en lenguaje ensamblador, o sea, binario, octal, decimal o hexadecimal.

Una vez tenemos el código fuente, podemos ensamblarlo, en dos pasos, para producir el código objeto.

- En ensambladores más potentes, normalmente con ordenadores de mayor tamaño, el fichero de código objeto se combina con otros ficheros para generar el código máquina, y en ensambladores más sencillos, este constituye directamente el propio código máquina, que es el ejecutable por la CPU.

En el primer paso se comprueban errores de sintaxis, errores de organización de memoria, y se calculan el espacio necesario y los desplazamientos de las direcciones relativas.

En el segundo paso, si no ha habido errores, se cumplimenta el código objeto, chequeando que los valores de los operandos estén en su rango, y las etiquetas estén en su lugar correcto (no haya etiquetas repetidas o inexistentes).

LD (IY+d),r

El contenido de cualquier registro r es transferido a la dirección de memoria especificada por la suma del contenido del par IY y el desplazamiento d (d es un número de 8 bits en complemento a 2).

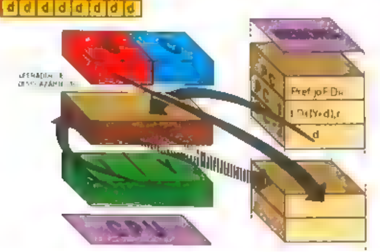
Mnemónico: LD **Operandos:** (IY+d),r

Formato binario:

1 1 1 1 1 1 0 1
0 1 1 1 0 r r r
d d d d d d d d

Ciclos: 5
Estados: 19 (4,4,3,5,3)

Indicadores: ninguno



LD r,(IY+d)

El contenido de la dirección de memoria especificada por la suma del contenido del par IY y el desplazamiento d (d es un número de 8 bits en complemento a 2), es transferido a cualquier registro r.

Mnemónico: LD **Operandos:** r,(IY+d)

Formato binario:

1 1 1 1 1 1 0 1
0 1 r r r 1 1 0
d d d d d d d d

Ciclos: 5
Estados: 19 (4,4,3,5,3)

Indicadores: ninguno

INSTRUCCIONES DE CARGA

Las instrucciones de carga transfieren contenidos de memoria a registros, de registros a memoria y entre registros.

Se trata del grupo principal de instrucciones del microprocesador, y su necesidad queda justificada, ya que todas las operaciones aritméticas y lógicas se hacen sobre registros del microprocesador, o entre estos y posiciones de memoria y casi siempre será necesario almacenar los resultados sobre la memoria

Por otra parte, gran número de instrucciones utilizan registros para direccionar posiciones de memoria, bien sea mediante direccionamiento absoluto o indexado.

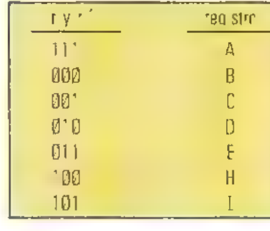
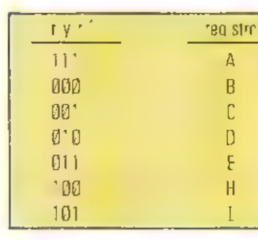
El formato básico de estas instrucciones es:

LD DESTINO, ORIGEN

El código LD del inglés "LOAD" (carga), indica al microprocesador que debe cargar en el "DESTINO" el valor contenido en el "ORIGEN".

El "DESTINO" y el "ORIGEN", pueden ser tanto registros, como posiciones de memoria, utilizaremos "r" y "r'" para referirnos a los registros de 8 bits, afectados por la instrucción, y "dd" para referirnos a los de 16 bits (pares de registros).

Los valores de "r" y "r'" usados para el código de máquina en este grupo de instrucciones, son los siguientes:

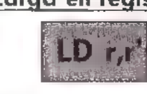



r y r'	registro
111	A
000	B
001	C
010	D
011	E
100	H
101	L

Los valores de "dd" usados para el código de máquina en este grupo de instrucciones, son los siguientes

dd	par de registros
00	BC
01	DE
10	HL
11	SP

Grupo de instrucciones de carga en registros




LD r,r'

OBJETO:

Carga el contenido del registro indicado por r', en el registro indicado por r.

CODIGO MAQUINA

01 <--r--><--r'-->

INDICADORES DE CONDICION A LOS QUE AFECTA

Ninguno

CICLOS DE MEMORIA·

1

CICLOS DE RELOJ:

4

EJEMPLO:

LD A,B

El contenido de "A" no es significativo, ya que será destruido por la instrucción. Supongamos que el contenido de "B" es 43 en decimal, 2Bh en Hexa.

(B) 00101011 2Bh

Ejecutamos la instrucción: LD A,B que carga en el registro "A", el contenido del registro "B":

LD A,B 01111000 78h

Después de la ejecución, el registro "A" contendrá el valor que contenía el registro "B", mientras que el contenido de este último no se habrá modificado

Contenido de "A" después de la ejecución:

(A) 00101011 2Bh

Contenido de "B" después de la ejecución:

(B) 00101011 2Bh

Como vimos en un capítulo anterior, los registros cumplen, en código máquina, una

función similar a la de las variables en Basic, de forma que esta instrucción sería similar a la instrucción: LET A=B del Basic.

LD r,n

OBJETO:

Carga en el registro indicado por "r" el valor numérico "n" de 8 bits y en el rango de 0 a 255.

CODIGO MAQUINA:

00 <- r -> 110
<- n ->

INDICADORES DE CONDICION A LOS QUE AFECTA

Ninguno

CICLOS DE MEMORIA:

2

CICLOS DE RELOJ:

7

EJEMPLO:

LD A,47

Esta instrucción carga el valor 47 decimal (2FH Hexa) en el registro "A", el contenido anterior de este registro se pierde al ejecutarse la instrucción.

La mayoría de los ensambladores, permiten introducir los números, tanto en decimal como en Hexa. Concretamente, en el caso del GENS 3, esta instrucción se podría escribir también como:

LD A, # 2F

El signo "#" delante del número, indica a el ensamblador que se trata de un número hexadecimal.

Instrucción.

LD A,47: 00111110 3Eh
00101111 2Fh

Contenido de "A" después de la ejecución:

(A) 00101111 2Fh

El equivalente en Basic de esta instrucción, sería: LET A=47

LD r,(HL)

OBJETO:

Carga en el registro indicado por "r", el contenido del octeto de memoria cuya dirección es el valor del par de registros HL.

CODIGO MAQUINA·

01 <--r--> 110

INDICADORES DE CONDICION:

Ninguno

CICLOS DE MEMORIA·

2

CICLOS DE RELOJ:

7

EJEMPLO:

LD B, (HL)

Esta instrucción carga en el registro "B", el contenido de la posición de memoria cuya dirección es el contenido del par de registros "HL". En este caso, estamos usando el modo de direccionamiento indirecto para especificar el "ORIGEN".

Supongamos que el registro "HL" contiene el valor 5F47h (24391), el registro "H" contendrá 5Fh (95) y el registro "L" contendrá 47h (71); observe que 95x256+71=24391.

La posición de memoria cuyo contenido vamos a cargar, será por tanto, la 5F47h Supongamos que a su vez, esta posición de memoria contiene el número 55h (85). Veamos cómo se desarrollan los acontecimientos.

Contenido del par de registros "HL":

(H) 01011111 5Fh
(L) 01000111 47h

Contenido de la posición de memoria 5F47h:

(5F47h) 01010101 55h

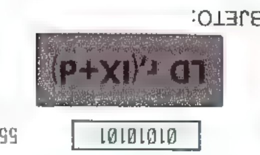
Ejecutamos la instrucción

LD B, (HL) 01000110 46h

Tras la instrucción, sólo se habrá modificado el contenido del registro "B".

Contenido del registro "B" después de la instrucción:

(B) 01010101 55h



LD r,(IX+d)

OBJETO:

Carga en el registro indicado por "r", el contenido de la posición de memoria, que resulta de sumar el valor del registro índice "IX" con un entero de desplazamiento "d", el cual puede adquirir los valores desde −128 a +127.

INDICADORES DE CONDICION:

Ninguno

CICLOS DE MEMORIA:

5

CICLOS DE RELOJ:

19

EJEMPLO:

LD (IY+10),B

Supongamos que el índice "IY" contiene 5F40h (24384), por lo que accederemos a la posición 5F4Ah (24394). Supongamos también, que el registro "B" contiene FFh (255). El contenido de la posición 5F4Ah no es significativo, ya que será destruido por la instrucción.

Contenido del índice "IY":

(IY) 01011111 5Fh
01000000 40h

Contenido del registro "B":

(B) 11111111 FFh

Ejecutamos la instrucción:

LD (IY+10),B 11111101 FDh
01110000 70h
00001010 0Ah

Contenido de la posición 5F4Ah después de la ejecución:

(5F4Ah)· 11111111 FFh

LD (HL),n

OBJETO:

Carga el valor del número entero "n", (entre 0 y 255) en la posición de memoria cuya dirección es el contenido del par de registros "HL"

CODIGO MAQUINA:

00110110
<- n ->

INDICADORES DE CONDICION:

Ninguno

CICLOS DE MEMORIA:

3

CICLOS DE RELOJ:

10

EJEMPLO:

LD (HL),57

Este ejemplo se podría escribir también como. LD (HL), # 39 ya que 39h = 57.

Supongamos que el par de registros "HL" contiene 6ACBh (27339), por tanto, esa será la dirección de memoria a la que accederemos. El contenido de esta posición de memoria no es significativo, ya que será destruido por la instrucción.

Contenido de "HL":

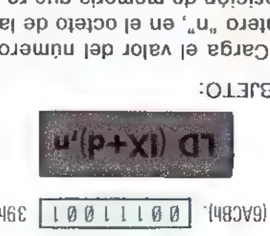
(HL)· 01101010 6A
11001011 CBh

Ejecutamos la instrucción:

LD (HL),57 00110110 36h
00111001 39h

Contenido de la posición 6ACBh después de la ejecución.

(6ACBh). 00111001 39h



LD (IX+d),n

OBJETO:

Carga el valor del número entero "n", en el octeto de la posición de memoria que resulta de sumar: el contenido del registro índice "IX" al entero de desplazamiento "d", el cual puede adquirir valores desde −128 a +127.

CODIGO MAQUINA:

11011101 DDh
00110110 36h
<- d ->
<- n ->

INDICADORES DE CONDICION:

Ninguno

CICLOS DE MAQUINA.

5

CICLOS DE RELOJ:

19

EJEMPLO:

LD (IX+3),7

Suponemos que "IX" contiene 73BCh (29628), por lo que accederemos a la posición 73BF (29631), cuyo contenido es irrelevante.

Contenido de "IX"

(IX) 01110011 73h
10111100 BCh

Ejecutamos la instrucción:

LD (IX+3),7 11011101 DDh
00110110 36h
00000011 03h
00000111 07h

Contenido de la posición 73BFh después de la ejecucion

(73BFh) 00000111

LD (IY+d),n

OBJETO:

Carga el valor del número entero "n", en el octeto de la posición de memoria que resulta de sumar el contenido del registro índice "IY" al entero de desplazamiento "d", el cual puede adquirir los valores desde −128 a +127.

CODIGO MAQUINA

11111101 FDh
00110110 36h
<- d ->
<- n ->

INDICADORES DE CONDICION.

Ninguno

CICLOS DE MEMORIA

5

CICLOS DE RELOJ.

19

EJEMPLO·

LD (IY+5),15

Le suponemos a "IY" un contenido de 5000h (20480), por lo que accederemos a la posición 5005h (20485). El contenido de esta posición es irrelevante.

Contenido de "IY":

(IY) 01010000 50h
00000000 00h

Ejecutamos la instrucción:

LD (IY+5),15 11111101 FDh
00110110 36h
00000101 05h
00001111 0Fh

Contenido de la posición 5005h después de la ejecución:

(5005h) 00001111 0Fh

Grupo de instrucciones de carga en registro acumulador

LD A,(BC)

OBJETO.

Carga en el registro acumulador, el contenido de la posición de memoria direccionada por el par de registros "BC".

CODIGO MAQUINA:

00001010 0Ah

INDICADORES DE CONDICION

Ninguno

CICLOS DE MEMORIA:

2

CICLOS DE RELOJ:

7

EJEMPLO:

LD A,(BC)

Supongamos que el par de registros "BC" contienen el número 76DFh (30431), esta es por tanto, la posición cuyo contenido cargaremos en el acumulador. Supongamos también, que el contenido de esta posición es AAh (170). El contenido del acumulador es irrelevante, ya que se pierde al ejecutar la instrucción.

Contenido del registro "BC".

(B) 01110110 76h
(C) 11011111 DFh

Contenido de la posición de memoria 76DFh.

(76DFh): 10101010 AAh

Ejecutamos la instrucción:

LD A,(BC) 00001010 0Ah

Contenido del acumulador después de la ejecucion

(A) 10101010 AAh

LD A,(DE)

OBJETO:

Carga en el registro acumulador, el contenido de la posición de memoria direccionada por el par de registros "DE"

CODIGO MAQUINA·

00011010 1Ah

INDICADORES DE CONDICION:

CODIGO MAQUINA:

1 1 0 1 1 1 0 1	DDh
0 1 < r > 1 1 0	
<------ ---d--------->	

INDICADORES DE CONDICION:

Ninguno

CICLOS DE MEMORIA:

5

CICLOS DE RELOJ:

19

EJEMPLO:

LD C,(IX+10)

En este caso, vamos a cargar el registro "C" con el contenido de la posición de memoria, cuya dirección es el resultado de sumar 10 al contenido del registro índice "IX".

Esta instrucción utiliza direccionamiento indexado para especificar el "ORIGEN"; obsérvese que el direccionamiento indexado es similar al indirecto, pero más sofisticado.

El contenido del registro "C" es irrelevante, ya que será destruido por la instrucción. Supongamos que el contenido de "IX" es 7743h (30531), por lo que accederemos a la posición de memoria 774Dh (30541). Supongamos también, que el contenido de esa posición de memoria es 41h (65).

Contenido de "IX":

(IX): 

77h
43h

Contenido de la posición de memoria 774Dh:

(774Dh): 0 1 0 0 0 0 0 1 41h

Ejecutamos la instrucción:

LD C,(IX+10):	11011101	DDh
	01001110	4Eh
	00001010	0Ah

Contenido de "C" después de la ejecución:

(C): 0 1 0 0 0 0 0 1 41h

Observe que la posición de memoria leída es 7743h+10, es decir 7743h+Ah=774Dh. Tanto el contenido de esta posición de memoria, como el del registro "IX", no han sido alterados.



OBJETO:

Carga en el registro indicado por "r", el contenido de la posición de memoria que resulta de sumar: el valor del registro índice "IY" con el entero de desplazamiento "d", el cual puede tomar valores desde –128 a +127.

CODIGO MAQUINA:



INDICADORES DE CONDICION:

Ninguno

CICLOS DE MEMORIA:

5

CICLOS DE RELOJ:

7

EJEMPLO:

LD A, (IY–15)

De forma similar al ejemplo anterior, vamos a cargar el acumulador con el contenido de la posición de memoria direccionada por el índice "IY" menos 15.

Supongamos que el contenido de "IY" es 7743h (30531), direccionamos, por tano la posición de memoria 7734h (30516), a la que a su vez, le suponemos un contenido de 42h (66).

(IY)	0 1 1 1 0 1 1 1	77h
	0 1 0 0 0 0 1 1	43h

Contenido de la posición de memoria 7734h:

(7734h): 0 1 0 0 0 0 1 0 42h

Ejecutamos la Instrucción:

LD A, (IY–15):	11111101	FDh
	01111110	7Eh
	11110001	F1h

Contenido de "A" después de la ejecución:

(A) 0 1 0 0 0 0 1 0 42h

Obsérvese que hemos representado –15 como F1h, que es precisamente el complemento a 2 de 0Fh es decir, el negativo de 15.

En el Z-80, el primer byte del código de operación de todas las instrucciones que utilizan el registro "IX" es DDh, y el de todas las que utilizan el "IY" es FDh.

Grupo de instrucciones de carga en memoria



OBJETO:

Carga en contenido del registro indicado por r, en el octeto de memoria direccionado por el valor del par de registros HL.

CODIGO MAQUINA:

0 1 1 1 0 <---r--->

INDICADORES DE CONDICION:

Ninguno

CICLOS DE MEMORIA:

2

CICLOS DE RELOJ:

7

EJEMPLO:

LD (HL),B

Esta instrucción carga en la posición de memoria cuya dirección es el contenido de "HL", el contenido del registro "B". Los contenidos previos de "B" y "HL" no son alterados y sí el contenido de la posición de memoria correspondiente.

En este caso, se utiliza direccionamiento indirecto para especificar el "DESTINO".

Supongamos que "HL" contiene 4723h (18211), ésta será por tanto, la posición a la que accederemos. Supongamos asimismo, que el registro "B" tiene un contenido de 75h (117). El contenido de la posición de memoria 4723 es irrelevante, ya que será destruido por la instrucción.

Contenido del par "HL":

(H):
(L): 

47h
23h

Contenido de "B":

(B): 

75h

Ejecutamos la instrucción:

LD (HL),B: 0 1 1 1 0 0 0 0 70h

Contenido de la posición 4723h después de la ejecución:

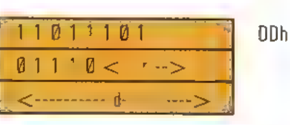
(4723h): 0 1 1 1 0 1 0 1 75h

LD (IX+d),r

OBJETO:

Carga el contenido del registro indicado por "r", en el octeto de la posición de memoria que resulta de sumar: el valor del registro índice "IX" con el entero de desplazamiento "d", el cual puede adquirir los valores desde –128 a +127.

CODIGO MAQUINA:



INDICADORES DE CONDICION:

Ninguno

CICLOS DE MEMORIA:

5

CICLOS DE RELOJ:

19

EJEMPLO:

LD (IX+7),C

Supongamos que "IX" contiene 75B3h (30131), por lo que accederemos a la posición 75BAh (30138), cuyo contenido es irrelevante. Suponemos también, que "C" contiene F0h (240).

Contenido del par "IX":

(IX) 

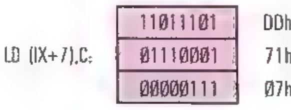
75h
B3h

Contenido de "C":

(C) 

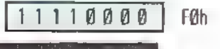
F0h

Ejecutamos la instrucción:

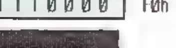
LD (IX+7),C: 

DDh
71h
07h

Contenido de la posición 75BAh después de la ejecución:

(75BAh): 

F0h




OBJETO:

Carga el contenido del registro indicado por "r", en el octeto de la posición de memoria resultante de sumar: el valor del registro índice "IY" al entero de desplazamiento "d" el cual puede adquirir los valores desde –128 a +127.

CODIGO MAQUINA:



GRAN CONCURSO MASTER-MIND

Después de llevarse a cabo el primer enfrentamiento en el Concurso Master-Mind (como anunciábamos en el número 42 de MICROHOBBY), pasamos a ofreceros la relación de clasificados para la segunda eliminatoria, que tendrá lugar los días 21, 22 y 23 de este mes de octubre, en el lugar, día y hora que abajo reseñamos.

UNA INICIATIVA DE



MICROHOBBY *SEMANAL*



Tienda: Modesto Lafuente, 63

Día	Hora	Concursante	
21/10	17,00	296-9	Luis E., Juan Enrique – J. Antonio Muñoz López
.	17,30	2-267	Juan Carmona – Juan A. García Navarro
.	18,00	3-268	Luis Miguel Espino - Francisco Menéndez
.	18,30	293 28	Arturo Lobo Gómez – Luis M. Brugarolas Martínez
.	19,00	5-269	José Luis Bueno Castilla – José Luis González Molinello
.	19,30	6-270	Manuel Cruz Brazales - Fausto Galdiano
22/10	17,00	7-26	J. Antonio Rodríguez Quintana – Rafael Luna Pérez
.	17,30	290-272	Miguel Angel Zurita – Andrés Muraya Díaz
.	18,00	289-12	José Antonio García - Manuel Minerva Quintero
.	18,30	287-273	J. J. Ibáñez – Fermín Trueba Hita
.	19,00	286-274	Emilio Mera Díaz – J. Fernando Brid
.	19,30	285-22	Arturo Lobo Gómez – Alfonso García Patiño
23/10	17,00	284-21	José Ferrer – Jesús Sancho Pastor
.	17,30	283-277	O. Sempere – José de la Riva Frías
.	18,00	282-20	Guillermo Cano – Miguel Samplón Chalmeta
.	18,30	16-19.	Jesús Castejón García – Sabino Samplón Chalmeta
.	19,00	17-10	Fernando Pardo Genovés – Francisco Soto Espinosa

Tienda: Colombia, 39-41

Día	Hora	Concursante	
21/10	17,00	237-89	Xavier García Faura – Javier Valdés Quirós
.	17,30	236-209	Angel España González - César M. Vicente Vilaseca
.	18,00	62 87	Manuel Cuadrado Villar – J. Enrique Cabellos Olmos
.	18,30	234-86	Gonzalo Ares – Miguel A. Zaplana Carreño
.	19,00	233 85	Carlos Granados Martínez – V. Solís
.	19,30	232-84	Antonio Noguero – Alfredo Bermúdez de Castro
22/10	17,00	66-82	Maite Muñoz – Francisco Mozo Villapún
.	17,30	230-216	Juan Polaino López – Angel González Valdenebro
.	18,00	229-218	R. Fernando Rada – Antonio Olivers García
.	18,30	69-78	Miguel Sesma Chércoler – Luis Arturo Ramos
.	19,00	227-224	José I. Marín – Alberto Martín Olano
.	19,30	226-223	Javier Hernández Ramos – Eugenio de Sancho
23/10	17,00	76-220	Luis Alvarez Satorren – Rafael Ferriz Martín
.	17,30	222-44	Juan Pujol – Esteban Esteban

Tienda: Ortega y Gasset, 21

Día	Hora	Concursante	
21/10	17,00	31-242	Juan Carmona Mercate – Manuel Albert Segarra
.	17,30	266-241	Luis Gala Pérez – Jaime Martín Roig
.	18,00	32-239	Carmelo García Redondo – José A. Bedía Domínguez
.	18,30	265-59	Alfredo Muñoz Alvarez – Juan Lorente Salinas
.	19,00	33-57	Vicente Jesús Roig Ros – Joaquín López
.	19,30	34-54	Pedro Surroca Sala - Rafael de las Heras
22/10	17,00	262-249	Miguel Shagún – Camilo Cela Elizagarete
.	17,30	36-248	Mercedes Corpodore – Jesús Manuel Vázquez
.	18,00	39-245	Carlos del Cunillo – Xavier Melich Martra
.	18,30	40-244	Rafael Martín Masa – Javier Muñoz Andújar
.	19,00	256-51	César García – Manuel Ciprián
.	19,30	259-50	Manuel Llanos de Paz – Juan Romera Arroyo
23/10	17,00	260-48	Carlos Pantaleón – Jesús González Mol
.	17,30	42-255	Juan M. Couchoud – José A. Rodríguez Fonollos
.	18,00	45-254	José Luis Bueno - Urbano García Barrós
.	18,30	46-252	José Balaguer – Juan Martínez Casany
.	19,00	47-251	Fdo. García Romero – Miguel Sánchez Bustamante

Tienda: Fuencarral, 100

Día	Hora	Concursante	
21/10	17,00	178-150	Fernando Recuerdo – Antonio Moyano Mulas
.	17,30	179-149	J. Angel Sánchez Caso – Javier Delgado
.	18,00	122-148	J. L. Sánchez Schoch - J. M. A. Taviel
.	18,30	124-147	Luis Arocha Hernandez – Roberto Rueda Pfeiffer
.	19,00	123-153	Sistoriayo Flors Torre – José M. Garrido Romero
.	19,30	125-145	Javier González – Javier Martínez Zapata
22/10	17,00	174-165	Lomas del Marbella – Javier Valdés Quirós
.	17,30	126 144	Francisco Carmona Moreno Antonio Gros Cambronero
.	18,00	172-143	Jesús Angel Serrano – Antonio Juan Hortigo
.	18,30	171-158	Ignacio Ramón Ferrer – Pedro Góez del Olmo
.	19,00	129-141	J. Ramón Sánchez Marín – Javier Ruano Franco
.	19,30	130-139	Francisco López Mudarra – Pedro Víctor Gómez Ramírez
23/10	17,00	169-160	Ismael Jiménez Calvo – Antonio Hormigo Varo
.	17,30	162-138	Juan A. Gómez Fabiani – Manuel Marmierca Zurdo
.	18,00	167-140	Francisco José Tolín Goviena – Jesús y Jordi
.	18,30	134-142	Francisco J. Rández García – Jesús Martínez Rivas
.	19,00	135-163	Carlos A. Roldán – Olavo Palomo López

Tienda: Padre Damián, 18

Día	Hora	Concursante	
21/10	17,00	90-180	Francisco Cid – Denis Dureux Panzal
.	17,30	91-181	Jorge Longobardo Quintas – Ramón Biosca Teixedo
.	18,00	92-118	Miguel Suárez Patiño – Fdo. García Moreno
.	18,30	204-182	Francisco Parro Redondo – Santiago Vila Doncel
.	19,00	203-116	Gonzo Suárez – Juan J. Delgado
.	19,30	95-184	Emilio González González – Luis Lacosta
22/10	17,00	200-185	Jesús García Rodríguez – Angel España González
.	17,30	199-187	Roberto Rueda Pfeiffer - Pedro Surroca Sala
.	18,00	99-188	Tomás Baño Coscolla – José M. Novo Fernández
.	18,30	100-109	Eduardo Roldán – Plácido García García
.	19,00	102-191	Juan Jesús León Cobos – Emilio Sanabre Hernández
.	19,30	103-108	Javier A. Motán Carrera – Alberto Garrido
.	17,00	195-192	Koldos Santos Bedoya – Juan José Ibánez
.	17,30	105-107	Rafael Ferrando Antonio A. García
.	18,00	106-	Jerónimo Pelegrín Gómez

NOTA: Las cintas no clasificadas pueden recogerse en la Tienda donde han concursado, o bien mandando 200 ptas. en sellos se enviarán a sus autores.

LOGO:

Los gráficos de la «tortuga» (I)

El Logo, un lenguaje cada vez más popular por méritos propios, está desplazando al Basic en la enseñanza primaria. La razón de este cambio es lo que pretendemos mostrar con este artículo acercando este lenguaje al Spectrum mediante una implementación que llamamos «Gráficos de Tortuga» (eso sí, dejando claro que ésta no pretende ser una versión estándar del Logo).

¡Eh! No os asustéis... Precisamente, Logo sólo pretende ser muy sencillo, e imponer las menos restricciones posibles al usuario. Tanto es así, que se ha llegado a enseñar a niños que aún no sabían leer ni escribir, mediante teclas de colores y figuritas.

Nosotros hemos intentado, más bien, que cada cual aprenda a su ritmo, jugando y divirtiéndose. Es un poco Aprender a Aprender... aprendiendo. Aprendiendo a jugar.

Este lenguaje gráfico, procedimental y recursivo también posee, en su versión completa, tratamiento de listas. En ello está la potencia del Lisp, lenguaje preferido de la Inteligencia Artificial... Pero esa es otra historia. Lo último: Logo sólo se aprende, se entiende, utilizándolo, experimentando cuanto se nos ocurra. Y para los profesores: la Geometría Diferencial que usa la Tortuga puede ser un ejemplo gráfico excelente para entender el Análisis Matemático...

El PROGRAMA

Los principales objetivos han sido dos: mínima memoria y máxima velocidad, todo ello sin limitar excesivamente la potencia de la implementación. Difícil cuestión.

Memoria

Para ahorrarla, se ha dividido el Programa en dos partes: la primera, es la inicialización, es decir, aquello que sólo se ha de hacer una vez, y al principio del programa: colocar las subrutinas en código máquina, inicializar todas las variables posibles, fijar los colores. Y, para no perder las variables recién definidas, en vez de LOAD se hace MERGE, con lo que todas las líneas del programa cargador desaparecen bajo las del programa principal. Todas menos la 3ØØØ, que es la que hace el MERGE. Si existiera también la línea 3ØØØ en el segundo programa, daría un error, C NO SENSE IN BASIC tras efectuar el Merge. ¡Ojo con esto!

El programa principal está poblado por una infinidad de VAL. Esto se debe a que, por la forma de representar o guardar en memoria los números en el Spectrum, VAL "527" ocupa 3 bytes menos que 527. Y lo mismo reza para cualquier otro número. Pero funciona más lentamente, de ahí que en algunos sitios no se use. También se emplean las variables o,u,k,tir y si, con valores respectivos de 0,1,2,3 y 4. Ocupa 6 bytes menos la variable u que el número 1, con igual velocidad.

Velocidad

Se ha colocado al principio del programa principal las subrutinas más usadas o que han de ser más veloces, pues así sucede con las primeras líneas del programa: corren más velozmente. También por esa razón se han hecho esas líneas tan largas: menos líneas, más rapidez global. Y ocupa menos memoria. Para la elipse, se ha utilizado un método lógico, a partir de la trigonometría. Y, como una elipse de semiejes (radios) iguales es un círculo, en ese caso se usa CIRCLE. Para LLENA, que es simplemente una instrucción Fill, se usa un último recurso. Se cambia la dirección de inicio del programa mediante POKEs en la variable del sistema PROS. Con ello, se simula que el programa comienza en la línea 9ØØØ. De hecho, el listado sólo mostrará las líneas de ésta en adelante. Si tenemos un error, sólo es posible resolverlo con un GOTO 952Ø. El número máximo de dos parámetros también se ha adoptado por cuestión de velocidad. De hecho, el programa guarda las instrucciones con dos parámetros, siendo éstos la cadena vacía, caso de no tenerlos.



Funcionamiento

En la línea 61ØØ comienza la subrutina de ejecución. Esta llama a la 6ØØØ, que le devuelve una instrucción sintácticamente correcta. Entonces, según el código de ésta, se ejecuta de una manera u otra: llamando a la subrutina que ejecuta la primitiva en cuestión, o llamando a la subrutina que ejecuta definidas. Esta actúa de la misma manera, llamándose a sí misma cuando es necesario, hasta que se termina de ejecutar la definida. Todas las palabras se ejecutan según este esquema. En la línea 7ØØØ se encuentra la subrutina de compilación, que también hace uso de la subrutina de edición en línea 6ØØØ.

La subrutina de edición toma una instrucción de las que se le han introducido por teclado, la chequea y devuelve el resultado, cuando es correcto. Si no, da mensaje de error y considera que ya ha ejecutado toda la línea. Y así hasta terminar la línea, donde pide de otra nueva línea y recomienza.

Existen diferentes mensajes de error, todos ellos muy claros.

La ventana gráfica

La tortuga se mueve sobre los ejes coordenados x,y, sin ningún límite que nos pueda dar error. Así, podremos dibujar un cuadrado en la posición 123ØØØ,548355.4. El único defecto es que no saldrá nada en la pantalla. Sólo deja rastro al pasar por la «ventana» gráfica. Esta no es más que un pedazo de pantalla, definido por sus vértices inferior izquierdo (MIN) y superior derecho (MAX). En principio, éstos coinciden con los respectivos de la pantalla; es decir: MIN Ø Ø; MAX 255 175. Pero pueden alterarse sin problemas..., salvo que los situemos fuera de la propia pantalla. Además, la tortuga sólo deja rastro cuando lleva el lápiz bajo: BAJA, lo baja y SUBE, lo sube. Fácil. Tampoco PINTA, ELIPSE ni LLENA dan errores. ELIPSE sólo dibuja si cabe ésta en ventana, y PINTA sólo colorea el fragmento de ventana que cubra. Lo mejor es probar todo esto.

MUY IMPORTANTE: Ante cualquier error, GO TO LOGO. Nunca RUN.

Bibliografía sobre LOGO

Introducción al Logo, de Boris Allan. Edita: Díaz de Santos, S. A.

Primeros pasos en Logo, por Marie Gaelle Monteil. Paraninfo.

Logo, de Anne Sparrowhawk. Personal Computer News. Versión inglesa.

Aprendiendo con Logo, por Daniel Watt. Edita: MacGraw-Hill.

NOTA IMPORTANTE: *El listado que publicamos es el programa principal. No puede funcionar solo, ya que le falta la parte de inicialización de variables, que publicaremos la semana próxima.*

PROGRAMA 2

```
   1 LET x12345678901=p2  RETURN
 . DEF FN c(x,y)=(x>=nx AND x<=mx AND y>=ny AND y<=my)  DEF FN d(w)=(w=o OR w=u OR w=k)
   2 IF ang>PI+PI OR ang<-PI-PI THEN LET ang=ang-2*PI*INT (ang/2/PI): LET sa=SIN ang· LET ca=COS ang: RETURN
   3 LET sa=SIN ang: LET ca=COS ang. RETURN
  10 LET cod=CODE p$(n). LET p1=CODE p$(n+u): LET p2=CODE p$(n+k)+n+p1+k; LET a$=p$(n+tir TO n+k+p1): LET b$=p$(n+tir+p1 TO p2)· LET n=p2+u: GO SUB 20+(10*cod AND cod<defcod): IF n<fin AND INKEY$<>" STOP " THEN GO TO 10
  11 RETURN
  12 FOR n=u TO maxcod. IF d$<>n$(n,k TO ) THEN NEXT n: LET n=256
  13 LET cod=n: RETURN
  16 IF CODE i$=32 THEN LET i$=i$(k TO )  GO TO 16
  17 RETURN
  18 FOR n=u TO LEN i$: IF i$(n)<>" " THEN NEXT n
  19 RETURN
  20 IF CODE n$(cod) THEN LET p1=VAL a$  IF CODE n$(cod)=k THEN LET p2=VAL b$
  21 IF n>=fin THEN LET n=l(cod)  LET fin=l(cod+u)-u. LET a=p1· LET b=p2  RETURN
  22 IF n<fin THEN LET p(punt)=a: LET p(punt+u)=b  LET p(punt+k)=n  LET p(punt+tir)=fin  LET punt=punt+si
  23 LET n=l(cod)  LET fin=l(cod+u)-u  LET a=p1  LET b=p2  GO SUB 10· LET punt=punt-si  LET a=p(punt)· LET b=p(punt+u)  LET n=p(punt+k)  LET fin=p(punt+tir)  RETURN
  24 PLOT x,y  LET x=x+(IN 49150<>k AND x<mx)-(IN 65022<>k AND x>nx)  LET y=y+(IN 56318<>k AND y<my)-(IN 32510<>k AND y>ny)  PLOT x,y  IF IN 59390=k THEN GO TO 24
  25 LET o=IN VAL "63486": IF k=o THEN LET k=u+u· LET o=NOT u: LET xx=x: LET yy=y  RETURN
  26 IF o>VAL "188" THEN OVER o=VAL "189"  GO TO VAL "24"
  27 IF o>VAL "175" THEN INVERSE o=VAL "183": GO TO VAL "24"
  28 PLOT OVER u,x,y: PLOT OVER LN EXP u;x,y: IF IN VAL "6398"=k THEN GO TO VAL "28"
  29 GO TO VAL "24"
  30 LET n=fin: RETURN
  40 IF punt<248 THEN LET p(punt)=lin. LET p(punt+u)=i. LET punt=punt+k: LET lin=n· LET i=VAL a$  RETURN
  41 LET p2=tir: GO TO error
  50 LET i=i-u: IF i>o THEN LET n=lin. RETURN
  51 LET punt=punt-k: LET lin=p(punt): LET i=p(punt+u): RETURN
  60 IF VAL a$ THEN RETURN
  61 LET n=n+CODE p$(n+u)+CODE p$(n+k)+tir: RETURN
  70 LET ca=PEEK VAL "23624"  LET sa=PEEK VAL "23693": LET maxcod=maxcod+u. GO SUB VAL "7000"· LET a$="1"  LET b$="0"  GO SUB VAL "140"  POKE VAL "23693",sa  POKE VAL "23624",ca  LET ca=COS ang  LET sa=SIN ang  LET p1=LEN l$  LET p$(final TO final+p1)=l$  LET final=final+p1· LET l(i+u)=final  RETURN
  80 LET p1=VAL a$: LET x=xx+ca*p1  LET y=yy+sa*p1  IF NOT raya THEN LET xx=x: LET yy=y: RETURN
  81 IF FN c(x,y) AND FN c(xx,yy) THEN PLOT xx,yy  DRAW x-xx,y-yy. LET xx=x: LET yy=y. RETURN
  82 LET x=xx· LET y=yy: FOR w=u TO p1  LET x=x+ca  LET y=y+sa  IF FN c(x,y) THEN PLOT x,y  NEXT w
  83 NEXT w
  84 LET xx=x· LET yy=y: RETURN
  85 LET x=(x3)*d/(z3+d)+ox: LET y=(y3)*d/(z3+d)+oy: LET ca=o: LET sa=SGN (x-xx)
  86 IF x<>xx THEN LET w=ATN (y-yy)/(x-xx)+(PI AND SGN (x-xx)=-u)  LET ca=SIN w. LET ca=COS w
  87 IF raya THEN GO SUB 81. LET sa=SIN ang  LET ca=COS ang  RETURN
  88 LET xx=x: LET yy=y: RETURN
  90 LET ang=ang-VAL a$*modo  GO TO k
 100 LET ang=ang+VAL a$*modo: GO TO k
 110 LET ang=VAL a$*modo: GO TO k
 120 LET x=VAL a$  LET y=VAL b$  LET xx=x  LET yy=y  RETURN
 130 LET a$=(a$+"            ")( TO 12)  FOR w=u TO VAL "12"  POKE dir+w,CODE a$(w)  NEXT w. LET p2=VAL b$  GO TO u
 140 LET p1=VAL a$. LET p2=VAL b$  IF NOT (FN d(p1) AND FN d(p2)) THEN LET p2=tir  GO TO error
 141 POKE VAL "51003",VAL "64+(137 AND P1)+(27 AND P1=k)"  POKE VAL "51006",VAL "64+(137 AND P2)+(27 AND P2=k)": RANDOMIZE USR VAL "51001"  RETURN
 150 LET p1=VAL a$: LET p2=VAL b$: IF NOT (FN d(p1) AND FN d(p2)) THEN LET p2=tir  GO TO error
 151 POKE VAL "51015",VAL "64+(1
```

```
37 AND P1)+(27 AND P1=k)"  POKE
VAL "51018",VAL "64+(137 AND P2)
+(27 AND P2=k)". RANDOMIZE USR V
AL "51013"· RETURN
 160 LET p1=VAL a$. LET p2=VAL b
$  IF NOT (FN d(p1) AND FN d(p2)
) THEN LET p2=tir  GO TO error
 161 POKE VAL "51038",VAL "64+(1
37 AND P2)+(27 AND p2=k)"  POKE
VAL "51041",VAL "64+(137 AND P1)
+(27 AND P1=k)": RANDOMIZE USR V
AL "51036"  RETURN
 170 LET p1=VAL a$  IF p1<o OR p
1>VAL "9" THEN LET p2=tir: GO TO
 error
 171 INK p1· RETURN
 180 LET p1=VAL a$  IF p1<o OR p
1>VAL "9" THEN LET p2=tir: GO TO
 error
 181 PAPER p1: RETURN
 190 LET p1=VAL a$: IF p1<>o AND
 p1<>u AND p1<>VAL "8" THEN LET
p2=tir  GO TO error
 191 FLASH p1. RETURN
 200 LET p1=VAL a$  IF p1<>o AND
 p1<>u AND p1<>VAL "8" THEN LET
p2=tir  GO TO error
 201 BRIGHT p1  RETURN
 210 OVER NOT NOT VAL a$: RETURN

 220 INVERSE NOT NOT VAL a$. RET
URN
 230 LET p1=VAL a$  IF p1<o OR p
1>VAL "7" THEN LET p2=tir. GO TO
 error
 231 BORDER p1· RETURN
 240 LET p1=VAL a$  LET p2=VAL b
$  IF VAL "p1<0 OR p1>10 OR p2<-
60 OR p2>60" THEN LET p2=tir: GO
 TO error
 241 BEEP p1,p2  RETURN
 250 LET p1=VAL a$  IF NOT p1 TH
EN LET p2=tir. GO TO error
 251 LET modo=k*PI/p1· RETURN
 260 LET p1=VAL a$  LET p2=VAL b
$  IF p1<=o OR p2<=o THEN LET p2
=tir  GO TO error
 261 LET x=xx  LET y=yy. IF NOT
raya THEN RETURN
 262 IF p1+x>mx OR x-p1<nx OR y+
p2>my OR y-p2<ny THEN RETURN
 263 IF p1=p2 THEN CIRCLE x,y;p1
  RETURN
 264 FOR w=o TO PI/si+.001 STEP
u/SQR (p1*p1+p2*p2)  LET sa=SIN
w*p2  LET ca=COS w*p1. PLOT x+ca
,y+sa  PLOT x+ca,y-sa  PLOT x-ca
,y+sa  PLOT x-ca,y-sa  LET cod=s
a  LET sa=ca/p1*p2  LET ca=cod/p
2*p1  PLOT x+ca,y+sa· PLOT x+ca,
y-sa  PLOT x-ca,y+sa  PLOT x-ca,
y-sa  NEXT w  LET sa=SIN ang. LE
T ca=COS ang  RETURN
 270 IF NOT FN c(xx,yy) THEN RET
URN
 271 LET k=VAL "191". GO TO VAL
"24"
 280 LET p1=VAL a$  LET x=xx+ca*
p1  LET y=yy+sa*p1
 281 IF raya AND FN c(x,y) AND F
N c(xx,yy) THEN PLOT xx,yy: DRAW
 x-xx,y-yy,VAL b$
 282 LET xx=x· LET yy=y  RETURN
 290 PRINT a$;. RETURN
 300 LET p1=VAL a$. IF (p1<o OR
p1>su) OR p1<>INT p1 THEN LET p2
=tir  GO TO error
 301 IF p1=o THEN LOAD b$SCREEN$

 302 IF p1=u THEN LOAD b$CODE VA
L "51456",VAL "6912"
 303 IF p1=k THEN LOAD b$CODE VA
L "58368",VAL "6912"
 304 IF p1=tir THEN LOAD b$CODE
USR "a",VAL "168"
 305 IF p1=k+k THEN LOAD b$ DATA
 p$()  LOAD b$ DATA n$()  LOAD b
$ DATA l()  LET maxcod=l(VAL "25
7")  LET final=l(VAL "258")
 306 RETURN
 310 LET p1=VAL a$  IF (p1<o OR
p1>si) OR p1<>INT p1 THEN LET p2
=tir  GO TO error
 311 IF p1=o THEN VERIFY b$SCREE
N$
 312 IF p1=u THEN VERIFY b$CODE
VAL "51456",VAL "6912"
 313 IF p1=k THEN VERIFY b$CODE
VAL "58368",VAL "6912"
 314 IF p1=tir THEN VERIFY b$COD
E USR "a",VAL "168"
 315 IF p1=si THEN VERIFY b$ DAT
A p$()  VERIFY b$ DATA n$(). VER
IFY b$ DATA l()
 316 RETURN
 320 LET p1=VAL a$  IF (p1<o OR
p1>si) OR p1<>INT p1 THEN LET p2
=tir· GO TO error
 321 IF p1=o THEN SAVE b$SCREEN$

 322 IF p1=u THEN SAVE b$CODE VA
L "51456",VAL "6912"
 323 IF p1=k THEN SAVE b$CODE VA
L "58368",VAL "6912"
 324 IF p1=tir THEN SAVE b$CODE
USR "a",VAL "168"
 325 IF p1=si THEN LET l(VAL "25
7")=maxcod  LET l(VAL "258")=fin
al· SAVE b$ DATA p$()  SAVE b$ D
ATA n$()  SAVE b$ DATA l()
 326 RETURN
 330 GO TO VAL a$
 340 LET p1=VAL a$· IF VAL "p1<-
65535 OR p1>65535" THEN LET p2=t
ir  GO TO error
 341 LET p1=USR p1  RETURN
 350 LET raya=o  RETURN
 360 LET raya=u  RETURN
 370 LET raya=u  LET x=VAL "128"
  LET y=VAL "88": LET xx=x: LET
yy=y· LET ang=o: LET sa=o: LET c
a=u  RETURN
 380 RETURN
 390 LET d$=a$: IF a$="ULT" OR a
$="ult" THEN LET d$=n$(maxcod,k
TO )
 392 GO SUB VAL "12": IF cod=VAL
 "256" THEN LET p2=VAL "5". LET
cod=VAL "37"· GO TO error
 393 IF cod<defcod THEN LET p2=t
ir· GO TO error
 394 PRINT INVERSE u;" LISTADO D
E ",d$,"       ",CODE n$(cod),TAB
 CODE " "
 395 FOR w=l(cod) TO l(cod+u)-u·
 LET p1=CODE p$(w+u)  LET p2=COD
E p$(w+k)+p1+w+k· PRINT n$(CODE
p$(w),k TO )," ";p$(w+tir TO p1+
w+k)," ";p$(w+tir+p1 TO p2): LET
 w=p2· NEXT w: RETURN
 400 IF maxcod<defcod THEN LET p
2=tir: GO TO error
 401 LET d$=a$: IF a$="ULT" OR a
$="ult" THEN LET d$=n$(maxcod,ti
r TO )
 402 IF a$="DEFS" OR a$="defs" T
HEN LET d$=n$(defcod,k TO )
 404 GO SUB VAL "12"  IF cod=VAL
 "256" THEN LET p2=VAL "5". LET
cod=VAL "38". GO TO error
 405 LET maxcod=cod-u· LET final
=l(cod)  RETURN
 410 CLS : RETURN
 420 COPY   RETURN
 430 LET d$=a$  IF a$="TODO" OR
a$="todo" THEN LET d$="ALTO"
 431 IF a$="DEFS" OR a$="defs" T
HEN LET d$=n$(defcod,k TO )
 432 GO SUB VAL "12". IF cod=VAL
 "256" THEN LET p2=VAL "5". LET
cod=VAL "41". GO TO error
 433 FOR w=cod TO maxcod  PRINT
" " AND w<VAL "100"," " AND w<VA
L "10";w;"   ",n$(w,k TO ),CODE
n$(w). NEXT w· RETURN
 440 PRINT VAL a$;: RETURN
 450 LET p1=VAL a$  LET p2=VAL b
$· PRINT AT VAL "p1-22*INT (p1/2
2)",VAL "p2-32*INT (p2/32)";: RE
TURN
 460 LET p1=VAL a$. LET p2=VAL b
$. LET x=xx+p1· LET y=yy+p2  LET
 p1=nx+(x-nx AND x>nx)+(mx-x AND
 x>mx): LET p2=ny+(y-ny AND y>ny
)+(my-y AND y>my)
 461 LET x=nx+(xx-nx AND xx>nx)+
(mx-xx AND xx>mx)· LET y=ny+(yy-
ny AND yy>ny)+(my-yy AND yy>my)
 462 LET p1=p1-x  FOR y=y TO p2
STEP SGN (p2-y)· PLOT x,y. DRAW
p1,o· NEXT y  RETURN
 470 IF VAL "USR 51056+USR 51110
"+PI THEN RETURN
 480 LET p1=VAL a$
 481 IF p1<=o THEN RETURN
 482 IF VAL "USR 3582'+PI THEN L
ET p1=p1-u· GO TO VAL "481"
 490 LET x=VAL a$. LET y=VAL b$
 491 GO SUB VAL "81"+(tir AND ra
ya)· RETURN
 500 LET a1=u  LET a2=o: LET a3=
o· LET b1=o. LET b2=u  LET b3=o.
 LET c1=o: LET c2=o. LET c3=u. R
ETURN
 510 LET p1=VAL a$  LET x3=x3+p1
*a1  LET y3=y3+p1*a2  LET z3=z3+
p1*a3  GO TO VAL "85"
 520 LET p1=VAL a$*modo  LET p2=
SIN p1· LET p1=COS p1  LET x=b1
 LET b1=x*p1+c1*p2. LET c1=c1*p1
-x*p2: LET x=b2  LET b2=x*p1+c2*
p2  LET c2=c2*p1-x*p2  LET x=b3·
 LET b3=x*p1+c3*p2. LET c3=c3*p1
-x*p2. LET x=xx: RETURN
 530 LET p1=VAL a$*modo. LET p2=
SIN p1· LET p1=COS p1. LET x=a1
 LET a1=x*p1+c1*p2  LET c1=c1*p1
-x*p2. LET x=a2  LET a2=x*p1+c2*
p2. LET c2=c2*p1-x*p2  LET x=a3:
 LET a3=x*p1+c3*p2. LET c3=c3*p1
-x*p2  LET x=xx. RETURN
 540 LET p1=VAL a$*modo: LET p2=
SIN p1: LET p1=COS p1  LET x=a1.
 LET a1=x*p1+b1*p2  LET b1=b1*p1
-x*p2· LET x=a2  LET a2=x*p1+b2*
p2  LET b2=b2*p1-x*p2. LET x=a3
 LET a3=x*p1+b3*p2  LET b3=b3*p1
-x*p2· LET x=xx. RETURN
 550 LET p1=VAL a$. IF p1<=o THE
N LET p2=tir. GO TO error
 551 LET d=p1  RETURN
 560 LET ox=VAL a$  LET oy=VAL b
$  LET p1=raya: LET raya=o. GO S
UB VAL "85": LET raya=p1: RETURN

 570 LET x3=VAL a$: RETURN
 580 LET y3=VAL a$. RETURN
 590 LET z3=VAL a$: RETURN
 600 GO TO VAL '85"
 610 IF NOT raya THEN RETURN
 611 LET p1=VAL a$  IF p1<>o AND
 p1<>u THEN LET p2=tir. GO TO er
ror
 612 GO TO VAL "8999"
 620 LET p2=VAL a$  IF p2<u OR p
2>VAL "8" THEN LET p2=tir
 621 IF p2=VAL "8" THEN LET i$=b
$  GO TO VAL "8020"
 622 GO TO error
 630 RETURN
 640 IF a$="ULT" OR a$="ult" THE
N LET a$=n$(maxcod,k TO )
 641 LET d$=a$: GO SUB VAL "12"
 IF cod<defcod OR cod=VAL "256"
THEN LET p2=VAL "52". GO TO erro
r
 642 LET sa=PEEK VAL "23624". LE
T ca=PEEK VAL "23693". LET i=cod
. LET sa=PEEK VAL "23693". BORDE
R VAL "4"  PAPER VAL "7"  INK o:
 CLS : PRINT INVERSE 1;" MODIFIC
AR   ";n$(i,k TO );"   ";CODE n
$(i),TAB CODE " ": GO SUB VAL "3
95"
 643 LET l$=p$(l(i) TO l(i+u)-u)
: GO SUB VAL "7100"
 644 LET p1=LEN l$  LET p$(l(i)
TO p1+final+si)=l$+p$(l(i+u) TO
final+si)
 645 LET p1=p1+l(i)-l(i+u). FOR
w=i+u TO maxcod+u: LET l(w)=l(w)
+p1: NEXT w
 646 LET final=final+p1. LET a$=
"1": LET b$="0". GO SUB VAL "140
"
 647 POKE VAL "23693",sa. POKE V
AL "23624",ca  LET ca=COS ang: L
ET sa=SIN ang  RETURN
 648 LET ca=PEEK VAL "23624": LE
T sa=PEEK VAL "23693". LET maxco
d=maxcod+u: GO SUB VAL "7000"  L
ET a$="1"  LET b$="0"  GO SUB VA
L "140"· POKE VAL "23693",sa. PO
KE VAL "23624",ca  LET ca=COS an
g  LET sa=SIN ang· LET p1=LEN l$
. LET p$(final TO final+p1)=l$.
LET final=final+p1: LET l(i+u)=f
inal· RETURN
 650 LET nx=VAL a$: LET ny=VAL b
$  RETURN
 660 LET mx=VAL a$. LET my=VAL b
$. RETURN
 670 LET x=xx  LET y=yy. LET w=t
ir  IF VAL "NOT (w+x>255 OR x-w<
0 OR y+w>175 OR y-w<0)" THEN CIR
CLE OVER u,x,y,w
 671 LET x=x+si*ca  LET y=y+si*s
a. IF VAL "u+x>255 OR x-u<0 OR y
+u>175 OR y-u<0" THEN RETURN
 672 CIRCLE OVER u;x,y,u· RETURN

 680 LET falso=NOT VAL a$. LET c
ierto=NOT falso  RETURN
 690 LET x=xx+VAL a$. LET y=yy+V
AL b$  GO TO VAL "81"
5999 REM  EDITOR
6000 GO SUB 16· IF i$="" THEN IN
PUT LINE i$· LET i$=i$+" ": GO T
O 6e3
6010 LET a$="": LET b$="". GO SU
B 18  LET d$=i$( TO n-u): LET i$
=i$(n TO ). GO SUB 16
6020 GO SUB 12. IF cod=256 THEN
LET p2=VAL "6"  GO SUB error· GO
 TO VAL "6e3"
6030 IF CODE n$(cod) THEN IF i$<
>"" THEN GO SUB 18  LET a$=i$( T
O n-u)  LET i$=i$(n TO ): GO SUB
 16
6040 IF CODE n$(cod)=k THEN IF i
$<>"" THEN GO SUB VAL "18". LET
b$=i$( TO n-u)· LET i$=i$(n TO )
  GO SUB VAL "16"
6050 LET par=(a$<>"")+(b$<>"")
IF par<>CODE n$(cod) THEN LET p2
=k· GO SUB error. GO TO VAL "6e3
"
6060 RETURN
6100 LET punt=u  GO SUB 6000
6110 IF cod=si OR cod=repe OR co
d=tir OR cod=u THEN LET p2=VAL "
7". GO SUB error: GO TO VAL "610
0"
6120 GO SUB 670  GO SUB 20+(10*c
od AND cod<defcod)· GO SUB 670
GO TO 6100
6130 GO SUB 20+(10*cod AND cod<d
efcod). GO TO 6100
7000 REM  COMPILACION
7010 LET l$=""  LET b=VAL b$· IF
 b<>o AND b<>u AND b<>k THEN LET
 p2=tir  GO TO error
7020 LET b$=i$  LET i$=a$  GO SU
B VAL '18'  LET a$=i$( TO n-u)
IF LEN a$>VAL '6" THEN LET p2=u
 GO TO error
7030 LET i$=b$  LET i=maxcod  LE
T n$(i)=CHR$ b+a$  LET a$="0'  L
ET b$="1"· GO SUB VAL "140"  PAP
ER VAL "7": INK o  BORDER VAL "5
". INVERSE o  FLASH o  BRIGHT o
 OVER o
7050 CLS · PRINT INVERSE u,"
 ",n$(i,k TO ),CODE n$(i),
7100 GO SUB 6e3  GO SUB 12: IF n
=5 THEN LET p2=VAL "7": GO SUB e
rror. GO TO VAL "7100"
7110 IF cod=36 THEN RETURN
7115 IF cod=61 THEN LET l$="". G
O TO VAL "7040"
7120 IF b>tope THEN LET p2=si: G
O SUB error  GO TO VAL "7100"
7140 PRINT d$," ",a$," ";b$
7150 LET p1=LEN a$  LET p2=LEN b
$: LET l$=l$+CHR$ cod+CHR$ p1+CH
R$ p2+a$+b$. LET b=b+3+p1+p2· GO
 TO 7100
8000 REM  ERROR
8010 RESTORE VAL "8100"  LET p1-
n· FOR n=u TO p2  READ i$  NEXT
n. NEXT n  LET n=p1
8020 LET p1=n: PRINT #u, INK VAL
 "7", PAPER o,i$," ",n$(cod,k TO
 )  LET i$=""  FOR n=VAL "-.1" T
O VAL ".1" STEP VAL ".02"· BEEP
ABS n,60-300*ABS n· NEXT n  LET
n=p1  PAUSE VAL "200": RETURN
8100 DATA "NOMBRE DEMASIADO LARG
O","No. PARAMETROS ERRONEO","NUM
ERO FUERA DE RANGO","NO QUEDA ME
MORIA DISPONIBLE"
8110 DATA "PALABRA NO DEFINIDA",
"MENSAJE NO INTERPRETABLE","USO
EN COMPILACION DE"
8999 LET f1=PEEK VAL "23637"· LE
T f2=PEEK VAL "23638"  GO TO VAL
 "9500"
9000 REM  FILL
9010 LET p2=x. LET si=o: LET sa=
u. IF y<my THEN LET sa=(p1=POINT
 (p2,y+u))
9014 LET ca=u· IF y>ny THEN LET
ca=(p1=POINT (p2,y-u))
9015 IF p1=POINT (p2,y) THEN RET
URN
9020 IF p1=POINT (p2-u,y) THEN G
O TO VAL "9025"
9021 LET p2=p2-u· IF p1=POINT (p
2,y-u) THEN LET si=u
9023 IF p2>nx THEN GO TO VAL "90
20"
9025 IF si AND p1<>POINT (p2,y-u
) THEN LET a$=CHR$ p2+CHR$ (y-u)
+a$. LET si=o
9030 IF y<my THEN IF (p1=POINT (
p2,y+u))<>sa THEN LET sa=NOT sa
 IF NOT sa THEN LET a$=CHR$ p2+C
HR$ (y+u)+a$
9040 IF y>ny THEN IF (p1=POINT (
p2,y-u))<>ca THEN LET ca=NOT ca·
 IF NOT ca THEN LET a$=CHR$ p2+C
HR$ (y-u)+a$
9050 PLOT INVERSE NOT p1,p2,y: L
ET p2=p2+u: IF p2<=mx THEN IF p1
<>POINT (p2,y) THEN GO TO VAL "9
020"
9060 RETURN
9500 LET w=PEEK VAL "23635"  POK
E VAL "23635",f1. LET f1=PEEK VA
L "23636": POKE VAL "23636",f2
LET f2=f1  LET f1=w: LET a$=""
LET x=xx  LET y=yy
9501 IF y>my THEN GO TO VAL "951
0"
9502 IF p1<>POINT (x,y+u) THEN L
ET y=y+u  GO TO VAL "9501"
9510 GO SUB VAL "9010"· LET y=y-
u. IF p1<>POINT (x,y) AND y>=ny
THEN GO TO VAL "9501"
9520 IF LEN a$ THEN LET x=CODE a
$. LET y=CODE a$(k)  LET a$=a$(t
ir TO )· GO TO VAL "9510"
9530 LET x=xx  LET y=yy  LET sa=
SIN ang. LET ca=COS ang  LET si=
k+k  POKE VAL "23635",f1  POKE V
AL "23636",f2· RETURN
9999 SAVE "Logo" LINE 6100. GO T
O 9999
```

EL PASTOR

José A. FERNANDEZ TOME

Spectrum 48 K

Un ambiente bucólico no siempre es el marco del pastor. Muy por el contrario, las inclemencias del tiempo y la vigilancia constante a su rebaño suelen ser los elementos principales de su jornada.

En esta ocasión, nuestro sufrido pastor ha perdido a seis de sus mejores ovejas. Esta situación se agraba considerablemente si tenemos en cuenta la oscuridad de la noche y la llegada de los lobos.

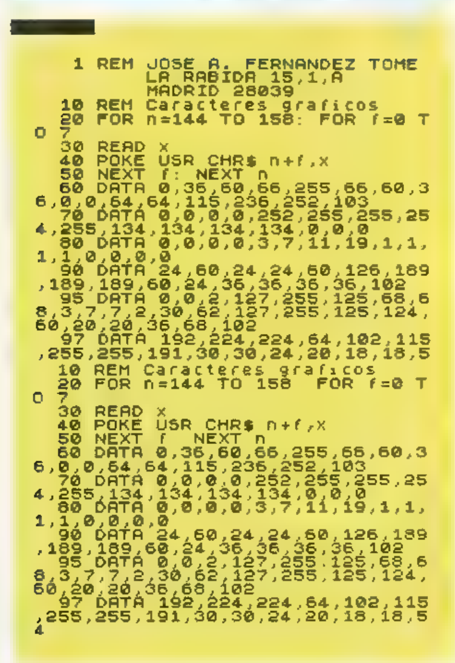
Nosotros, como protagonistas de esta angustiosa situación, hemos de intentar ir en su busca ayudándonos de bengalas para su localización siempre teniendo en cuenta que nos acechan tres lobos y que seis cepos esperan abrazar a la presa.

Premiado con 15.000 ptas.



NOTAS GRAFICAS

A B C D E F G H I J K L M N O



```
   1 REM JOSE A. FERNANDEZ TOME
         LA RABIDA 15,1,A
         MADRID 28039
  10 REM Caracteres graficos
  20 FOR n=144 TO 158: FOR f=0 T
O 7
  30 READ x
  40 POKE USR CHR$ n+f,x
  50 NEXT f: NEXT n
  60 DATA 0,36,60,66,255,66,60,3
6,0,0,64,64,115,236,252,103
  70 DATA 0,0,0,0,252,255,255,25
4,255,134,134,134,134,0,0,0
  80 DATA 0,0,0,0,3,7,11,19,1,1,
1,1,0,0,0,0
  90 DATA 24,60,24,24,60,126,189
,189,189,60,24,36,36,36,36,102
  95 DATA 0,0,2,127,255,125,68,6
8,3,7,7,2,30,62,127,255,125,124,
60,20,20,36,68,102
  97 DATA 192,224,224,64,102,115
,255,255,191,30,30,24,20,18,18,5
  10 REM Caracteres graficos
  20 FOR n=144 TO 158  FOR f=0 T
O 7
  30 READ x
  40 POKE USR CHR$ n+f,x
  50 NEXT f  NEXT n
  60 DATA 0,36,60,66,255,66,60,3
6,0,0,64,64,115,236,252,103
  70 DATA 0,0,0,0,252,255,255,25
4,255,134,134,134,134,0,0,0
  80 DATA 0,0,0,0,3,7,11,19,1,1,
1,1,0,0,0,0
  90 DATA 24,60,24,24,60,126,189
,189,189,60,24,36,36,36,36,102
  95 DATA 0,0,2,127,255,125,68,6
8,3,7,7,2,30,62,127,255,125,124,
60,20,20,36,68,102
  97 DATA 192,224,224,64,102,115
,255,255,191,30,30,24,20,18,18,5
4
  98 DATA 129,195,165,153,129,12
9,129,255,0,0,0,195,165,153,129,
255
 100 CLS   BORDER 3  PRINT AT 10
,10; PAPER 5; INK 0; FLASH 1,"EL
 PASTOR",AT 13,13;"POR",AT 16,2,
" DAVID FERNANDEZ LORY Y CIA."
PRINT #0,"PULSA UNA TECLA". PAUS
E 0  GO SUB 4000
 105 LET camp=0  LET seg=0  LET
jug=0
 110 INPUT "Nombre del pastor ?"
,a$  IF LEN a$>10 THEN GO TO 110
 120 LET jug=jug+1
 200 REM Cepo
 210 LET a1=INT (RND*16)+1
 215 LET b1=INT (RND*31)
 220 LET a2=INT (RND*16)+1
 225 LET b2=INT (RND*31)
 230 LET a3=INT (RND*16)+1
 235 LET b3=INT (RND*31)
 240 LET a4=INT (RND*16)+1
 245 LET b4=INT (RND*31)
 250 LET a5=INT (RND*16)+1
 255 LET b5=INT (RND*31)
 260 LET a6=INT (RND*16)+1
 265 LET b6=INT (RND*31)
 400 REM Oveja
 410 LET e1=INT (RND*17)
 415 LET f1=INT (RND*31)
 420 LET e2=INT (RND*17)
 425 LET f2=INT (RND*31)
 430 LET e3=INT (RND*17)
 435 LET f3=INT (RND*31)
 440 LET e4=INT (RND*17)
 445 LET f4=INT (RND*31)
 450 LET e5=INT (RND*17)
 455 LET f5=INT (RND*31)
 460 LET e6=INT (RND*17)
 465 LET f6=INT (RND*31)
 480 INPUT "Dificultad (1-3)";di
f. IF dif<1 OR dif>3 THEN GO TO
480
 500 LET x=16  LET y=6  LET w=0
 LET q=-1: LET ov=0: LET vi=6/di
f: LET punt=0: LET beng=6/dif: L
ET r=0
 505 FOR n=0 TO 17  PRINT AT n,0
, PAPER 3;"
               ": NEXT n. FOR f=19 T
O 21  PRINT AT f,0; PAPER 4,"
                                 "
NEXT f  GO TO 550
 510 IF beng<0 THEN LET beng=0
GO TO 600
 512 LET A=53. PAPER 1  FOR N=56
 TO 154 STEP 3  LET A=A+2  PLOT
INK 7,A,N  PLOT INK 1,A-4,N-6  B
EEP .01,560/A. NEXT N
 513 LET B=125. LET C=125. LET D
=125. LET E=125  LET F=125. LET
G=125. LET H=125: FOR N=164 TO 1
00 STEP -2: LET B=B-3: LET C=C-2
: LET D=D-1.4: LET F=F+1.4: LET
G=G+2  LET H=H+3  INK 7  BEEP .0
1,-25  PLOT B,N  PLOT C,N  PLOT
D,N  PLOT E,N  PLOT F,N. PLOT G,
N: PLOT H,N: NEXT N
 515 FOR n=0 TO 18  PRINT AT n,0
; PAPER 7;"
               ": NEXT n: PRINT PAPE
R 7; INK 0;AT a1,b1;CHR$ 144;AT
c1,d1;CHR$ 148,AT c1,d1+1;CHR$ 1
46;AT c1,d1+2,CHR$ 145;AT c1+1,d
1,CHR$ 149,AT c1+1,d1+1,CHR$ 147
;AT e1,f1,CHR$ 152
 517 IF ov=6 OR vi=0 THEN PRINT
AT x,y, PAPER 7, INK 0,CHR$ (150
+w),AT x+1,y,CHR$ (151+w)
 520 PRINT PAPER 7; INK 0,AT e2,
f2,CHR$ 152,AT e3,f3,CHR$ 152;AT
 e4,f4;CHR$ 152,AT e5,f5;CHR$ 15
2;AT e6,f6,CHR$ 152
 530 PRINT PAPER 7, INK 0,AT c2,
d2,CHR$ 148,AT c2,d2+1;CHR$ 146,
AT c2,d2+2,CHR$ 145,AT c2+1,d2,C
HR$ 149,AT c2+1,d2+1,CHR$ 147,AT
 c3,d3,CHR$ 148,AT c3,d3+1,CHR$
146,AT c3,d3+2,CHR$ 145,AT c3+1,
d3;CHR$ 149;AT c3+1,d3+1;CHR$ 14
7
 540 PRINT PAPER 7, INK 0;AT a2,
b2,CHR$ 144,AT a3,b3,CHR$ 144,AT
 a4,b4;CHR$ 144;AT a5,b5;CHR$ 14
4,AT a6,b6,CHR$ 144
 543 IF ov=6 OR vi=0 THEN PRINT
AT 20,30, PAPER 4;" "  PAUSE 400
  RETURN
 545 PRINT AT 20,30, PAPER 4," "
  FOR n=1 TO 6/dif  PAUSE 110  N
EXT n
 550 LET d2=INT (RND*12)+8
 551 LET c3=INT (RND*16)+1
 552 LET d3=INT (RND*10)+20
 553 LET c1=INT (RND*16)+1
 554 LET d1=INT (RND*10)
 555 LET c2=INT (RND*16)+1
 560 FOR n=0 TO 17  PRINT AT n,0
; PAPER 1,"
               "  NEXT n  PRINT AT x
,y, PAPER 1, INK 7,CHR$ (150+w),
AT x+1,y,CHR$ (151+w)
 565 PRINT AT 19,26, PAPER 4, IN
K 0;"BENG.";AT 20,28,beng;AT 19,
13;"PUNTOS",AT 20,15;punt,AT 19,
20,"VIDAS",AT 20,22,vi
 570 FOR n=0 TO 5  PRINT AT 18,n
, PAPER 4, INK 1,"M"  NEXT n  FO
R f=7 TO 31  PRINT AT 18,f, PAPE
R 4, INK 1,"M"  NEXT f  PRINT AT
 18,6, PAPER 4; INK 1,"N"
 580 FOR n=1 TO 30  PRINT AT 21,
n, PAPER 4, INK 0,"▄"  NEXT n
 590 PRINT AT 19,0, PAPER 4; INK
 0,"▌",AT 20,0,"▌",AT 21,0,"▙",A
T 19,12,"▌",AT 20,12,"▌",AT 21,1
2,"▙",AT 19,19,"▌",AT 20,19,"▌",
AT 21,19,"▙",AT 19,25,"▌",AT 20,
25,"▌",AT 21,25,"▙",AT 19,31,"▐"
,AT 20,31,"▐",AT 21,31,"▟"
 600 IF vi<=0 THEN LET vi=0
 601 PRINT AT 20,22, PAPER 4, IN
K 0,vi;AT 20,15;punt
 603 PRINT AT 18,6; PAPER 4; INK
 0;"N"
 605 IF x=16 AND y=6 AND w=3 THE
N LET q=q+2  LET ov=ov+1: LET pu
nt=punt+(10*dif): PRINT AT 20,15
; PAPER 4; INK 0;"   ";AT 20,15
,punt;AT 18,6," ";AT 20,q;CHR$ 1
52. FOR n=20 TO 19 STEP -.2: BEE
P .1,n: NEXT n: LET w=0
 607 IF vi<=0 THEN LET punt=punt
+(beng*10*dif). PRINT AT 20,15,
PAPER 4,"    ",AT 20,15, INK 0,p
unt  BEEP 1,-10  BEEP .75,-10. B
EEP .25,-10: BEEP 1,-10: BEEP .7
5,-7  BEEP .25,-8  BEEP .75,-8:
BEEP .25,-10  BEEP .75,-10. BEEP
 .25,-11  BEEP 1,-10: GO SUB 515
  GO TO 3500
 615 IF ov=6 THEN LET punt=punt+
(beng*10*dif)+(vi*10*dif)+(10*di
f): PRINT AT 20,15; PAPER 4; INK
 0;punt. BEEP .2,16: BEEP .2,17.
 BEEP .2,21: BEEP .2,19  BEEP .1
,17: BEEP .1,16. BEEP .2,14: BEE
P .2.7  BEEP .2,12  BEEP .2,14
BEEP .2,16  BEEP .1,17  BEEP .1,
16  BEEP .2,14  BEEP .2,16  BEEP
 .2,17. BEEP .1,19. BEEP .1,17.
BEEP .1,16  BEEP .1,12  BEEP .1,
16  BEEP .1,19  BEEP .4,24  GO S
UB 515. GO TO 3500
 650 IF INKEY$="" THEN GO TO 650
 655 IF INKEY$="b" OR INKEY$="B"
 THEN LET beng=beng-1. GO TO 510
 660 IF INKEY$="8" THEN LET y=y+
1. BEEP .01,y: GO TO 2000
 670 IF INKEY$="5" THEN LET y=y-
1  BEEP .01,y: GO TO 2100
 680 IF INKEY$="7" THEN LET x=x-
1  BEEP .01,x  GO TO 2200
 690 IF INKEY$="6" THEN LET x=x+
1  BEEP .01,x  GO TO 2300
1000 IF INKEY$<>"" THEN GO TO 60
0
2000 IF y>31 THEN LET y=31
2010 GO SUB 3000
2030 PRINT AT x,y-1, PAPER 1," "
,AT x+1,y-1," ",AT x,y, INK 7,CH
R$ (150+w),AT x+1,y,CHR$ (151+w)
2040 GO TO 600
2100 IF y<0 THEN LET y=0
2110 GO SUB 3000
2120 IF w=0 THEN LET r=0
2125 IF w=3 THEN LET r=2
2130 PRINT AT x,y+1, PAPER 1," "
;AT x+1,y+1;" ";AT x,y; INK 7;CH
R$ (150+w+r);AT x+1,y;CHR$ (151+
w+r)
2140 GO TO 600
2200 IF x<0 THEN LET x=0
2210 GO SUB 3000
2230 PRINT AT x+2,y; PAPER 1," "
,AT x,y, INK 7,CHR$ (150+w),AT x
+1,y,CHR$ (151+w)
2240 GO TO 600
2300 IF x>=17 THEN LET x=16
2310 GO SUB 3000
2330 PRINT AT x-1,y, PAPER 1," "
;AT x,y; INK 7;CHR$ (150+w),AT x
+1,y,CHR$ (151+w)
2340 GO TO 600
3000 REM Captura de ovejas
3010 IF x=e1 AND y=f1 THEN LET e
1=20  LET f1=30  GO TO 3090
3015 IF x=e2 AND y=f2 THEN LET e
2=20. LET f2=30  GO TO 3090
3020 IF x=e3 AND y=f3 THEN LET e
3=20  LET f3=30. GO TO 3090
3025 IF x=e4 AND y=f4 THEN LET e
4=20  LET f4=30  GO TO 3090
3030 IF x=e5 AND y=f5 THEN LET e
5=20  LET f5=30  GO TO 3090
3035 IF x=e6 AND y=f6 THEN LET e
6=20  LET f6=30  GO TO 3090
3080 GO TO 3110
3090 LET w=3  LET r=2  BEEP .01,
19  BEEP .01,16  BEEP .01,12  RE
TURN
3100 REM Caida en cepo
3110 IF x+1=a1 AND y=b1 THEN LET
 a1=20  LET b1=30  GO TO 3190
3115 IF x+1=a2 AND y=b2 THEN LET
 a2=20. LET b2=30  GO TO 3190
3120 IF x+1=a3 AND y=b3 THEN LET
 a3=20  LET b3=30  GO TO 3190
3125 IF x+1=a4 AND y=b4 THEN LET
 a4=20: LET b4=30  GO TO 3190
3130 IF x+1=a5 AND y=b5 THEN LET
 a5=20. LET b5=30  GO TO 3190
3135 IF x+1=a6 AND y=b6 THEN LET
 a6=20  LET b6=30  GO TO 3190
3180 GO TO 3210
3190 LET vi=vi-1  LET punt=punt-
(5*dif)  FOR f=-20 TO -23 STEP -
.5  BEEP .01,f  NEXT f  RETURN
3200 REM Captura por lobo
3210 IF x=c1 AND y=d1 THEN GO TO
 3290
3215 IF x=c2 AND y=d2+2 THEN GO
TO 3290
3220 IF x=c3 AND y=d3+2 THEN GO
TO 3290
3280 RETURN
3290 LET vi=vi-2. LET punt=punt-
(10*dif)  FOR f=-20 TO -26 STEP
-.1  BEEP .01,f  NEXT f: BEEP .1
,-30  FOR N=1 TO 20. PRINT AT 10
,10; FLASH 1, PAPER 7; INK 1;"EL
 LOBO"  PAUSE 10: NEXT N: PRINT
AT 10,10, PAPER 1;"       "  RET
URN
3500 REM Clasificacion
3505 PAPER 3: PAUSE 300. CLS
3510 IF jug=1 THEN PAUSE 100. LE
T c$=a$  LET camp=punt  LET d$=a
$  GO TO 3550
3520 IF punt>camp THEN LET seg=c
amp  LET d$=c$: LET c$=a$: LET c
amp=punt  GO TO 3550
3530 IF punt<=camp AND punt>seg
THEN LET d$=a$  LET seg=punt
3550 PRINT AT 6,24; PAPER 3, INK
 0,"PUNTOS",AT 8,2,"CAMPEON",AT
8,13,c$,AT 8,26,camp,AT 11,2,"SU
BCAMPEON",AT 11,13,d$,AT 11,26,s
eg,AT 14,13,"JUGADA",AT 14,26,ju
g
3600 PRINT AT 20,2; PAPER 3; INK
 0,"JUEGAS OTRA ?"
3610 IF INKEY$="" THEN GO TO 361
0
3620 IF INKEY$="S" OR INKEY$="s"
 THEN GO TO 110
3630 IF INKEY$="N" OR INKEY$="n"
 THEN CLS  GO TO 4050
3640 IF INKEY$<>"" THEN GO TO 36
10
4000 CLS  PRINT AT 10,2,"NECESI
TAS INSTRUCCIONES ?(S/N)"  PAUSE
 100  CLS
4010 IF INKEY$="" THEN GO TO 400
0
4020 IF INKEY$="S" OR INKEY$="s"
 THEN GO TO 4100
4030 IF INKEY$="N" OR INKEY$="n"
 THEN RETURN
4040 IF INKEY$<>"" THEN GO TO 40
00
4050 PRINT AT 10,6;"!!HASTA LA P
ROXIMA!!"  PAUSE 300  PAUSE 0  G
O TO 4050
4100 PRINT " El pastor ha perdid
o sus ove-  jas; es de noche y p
ara poder   verlas dispone de va
rias benga- las (de 1 a 6), segu
n el grado  de dificultad elegid
o. Para u-  sarlas pulsa 'B'."
4110 PRINT   PRINT " Una vez ilu
minado el campo, a- pareceran 6
ovejas,6 cepos y 3  lobos;si fal
ta alguna de las     ovejas, es p
orque se encuentra  en el mismo
lugar que un lobo o un cepo, per
o el pastor los dis-tingue, ya q
ue a las ovejas las coge con las
 manos, en los ceposcae por los
pies y el lobo le a-trapa por la
 cabeza."
4120 PRINT : PRINT " Las ovejas
y los cepos permane-cen inmovile
s mientras dura el  juego, pero
los lobos se mueven en la oscuri
dad."
4130 PRINT #0;"Pulsa una tecla"
 PAUSE 0
4140 CLS : PRINT " Hay tres nive
les de dificultad,que determinan
 una distinta pun-tuacion y dist
into numero de vi-das y bengalas
 y la duracion de estas."
4150 PRINT : PRINT " El movimien
to se realiza con    las teclas d
el cursor. Las ben- galas se dis
paran con 'B' en    cualquier mo
mento del juego."
4160 PRINT : PRINT " El juego te
rmina cuando se re- cogen las 6
ovejas, o cuando se acaban las v
idas, puntuando las vidas o beng
alas que resten."
4170 PRINT   PRINT " Las ovejas
se deben recoger de una en una,
pues las que se co- gen de mas,
se pierden en el ca-mino y no se
 pueden recuperar."
4180 PRINT #0,"Pulsa una tecla"
4200 PAUSE 0  RETURN
```

TOBOGAN

Antonio GONZALEZ FERNANDEZ

Spectrum 48 K

Premiado con 15.000 ptas.

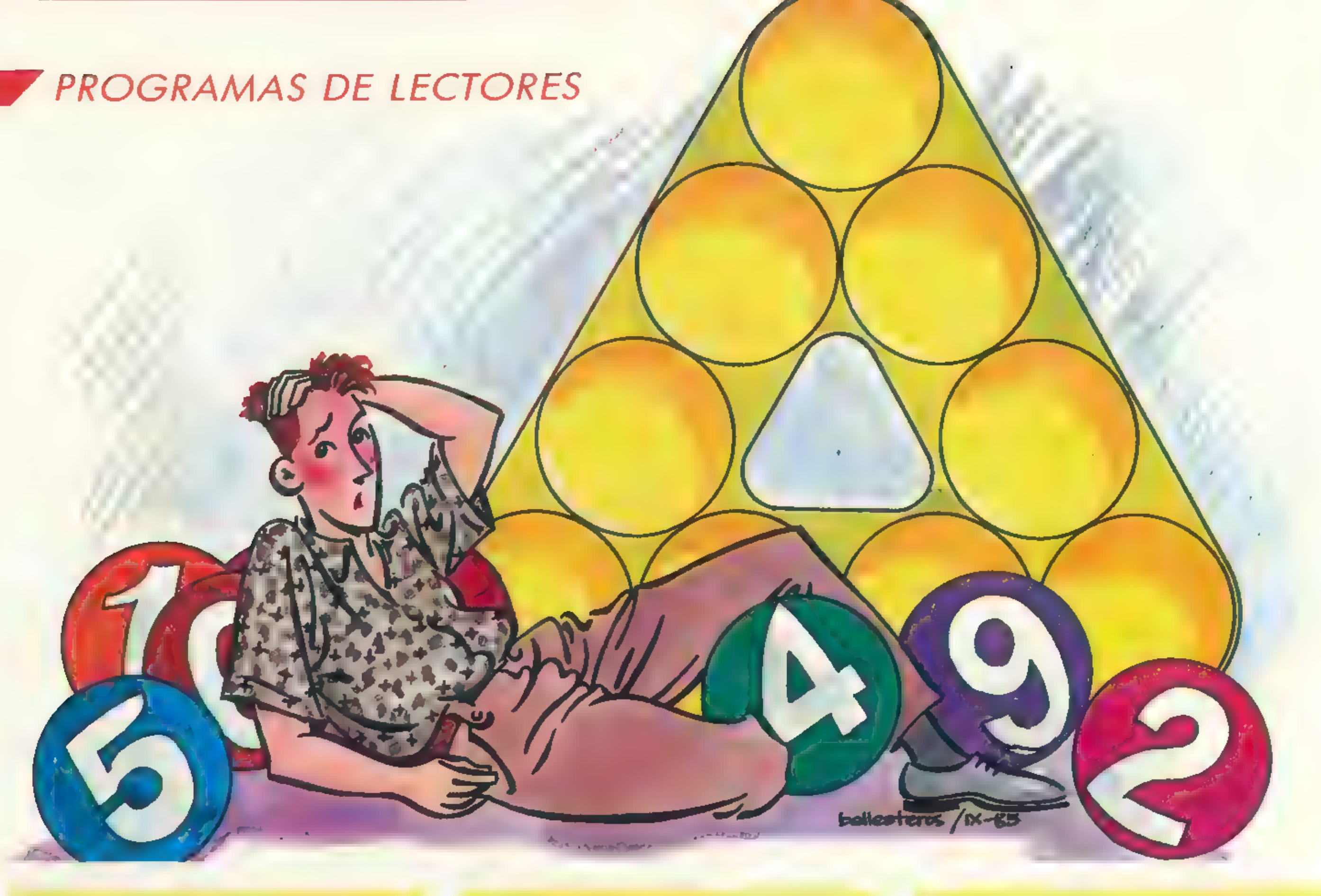


Con alguna similitud al famoso «Cubo de Rubik», este juego requiere, como aquél, habilidad y destreza (amén de paciencia).

Consiste en nueve círculos que forman un triángulo con el vértice hacia arriba, dentro de cada uno de los cuales va un número que debemos desplazar a nuestro antojo (bien en sentido horario o antihorario) hasta ordenarlos, desplazándose en cada movimiento cada número un lugar.

Con los ejemplos que incorpora el listado podréis comprobar que no es una tarea fácil.

```
   1 REM  ***********************
   2 REM  **     EL TOBOGAN    **
   3 REM  **   © A. Gonzalez   **
   4 REM  ***********************
   5
   6 REM  ***********************
   7 REM  **  INICIALIZACION   **
   8 REM  ***********************
  10 DIM C(9)  DIM F(9)  DIM N(9
)
  12 RANDOMIZE
  15 RESTORE
  17 LET VTOB=9500  LET PRINTOB=
5000  LET IMOV=6000: LET EMOV=70
00. LET F1=0: LET C1=1
  18 POKE 23658,8
  20 FOR I=1 TO 9
  30 READ C,F: LET F(I)=F. LET C
(I)=C
  40 NEXT I
  50 DATA 15,3,18,8,21,13,24,18,
18,18,12,18,6,18,9,13,12,8
  60 LET X=1  LET Y=4  LET Z=7
LET CONT=0
  98
  99 REM  ***********************
```



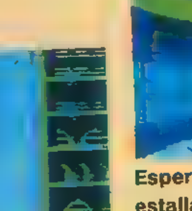

```
 100 REM  **   PRESENTACION   **
 101 REM  ***********************
 102
 110 BORDER 6. PAPER 6. INK 1. B
RIGHT 0. CLS
 120 PRINT AT 2,2,"ESTE ES EL JU
EGO DEL TOBOGAN";AT 4,2,"QUE DES
EA?";AT 6,3;"1 VER LAS REGLAS";A
T 8,3,"2 EXPERIMENTAR CON UN TOB
OGAN      ORDENADO",AT 11,3,'3 RE
SOLVER UN TOBOGAN",AT 13,3,'4 SA
LIR DEL PROGRAMA"
 130 IF INKEY$="" THEN GO TO 130
 135 IF INKEY$<>"1" AND INKEY$<>
"2" AND INKEY$<>"3" AND INKEY$<>
"4" THEN GO TO 130
 140 GO SUB 1000*VAL INKEY$
 150 GO TO 1
 998
 999 REM  ***********************
1000 REM  **    S/R REGLAS    **
1001 REM  ***********************
1002
1010 PAPER 7: INK 0: BORDER 7: C
LS
1020 FOR I=1 TO 9
1030 CIRCLE 2*C(I)+1,171-2*F(I)-
1,6
1040 NEXT I
1050 PRINT AT 1,7,'ESTE ES EL TO
BOGAN",AT 3,7,"DENTRO DE CADA CI
RCULO",AT 4,7,'IRA UN NUMERO, US
TED",AT 5,7,"DEBE ORDENAR LOS NU
MEROS";AT 6,0,"EN SENTIDO HORARI
O, CON EL 1 EN EL VERTICE SUPERI
OR'
1060 PRINT AT 9,0,"PARA ELLO PUE
DE MOVER UNA DE LASTRES HILERAS
 (X,Y o Z), DE FORMAQUE TODOS COR
REN UN LUGAR Y EL  ULTIMO SE PON
E EL PRIMERO, ESTE MOVIMIENTO PU
EDE SER EN SENTIDO HORARIO (-) O
 ANTIHORARIO (+)"
1070 PRINT AT 16,0;"ES ACONSEJAB
LE EXPERIMENTAR CON UN TOBOGAN A
RMADO PARA OBTENER  REGLAS GENER
ALES (OPCION 2) ANTES DE LANZARS
E A RESOLVER UNO"
1080 PRINT #1,"PULSE UNA TECLA P
ARA DEMOSTRA-  CION"
1090 PAUSE 0: CLS
1100 FOR I=1 TO 9
1120 READ N: LET N(I)=N
1130 NEXT I
1140 DATA 1,2,5,3,4,6,7,8,9
1200 GO SUB PRINTOB
1202 PRINT #1,AT 0,0,"PULSE UNA
TECLA PARA SEGUIR": PAUSE 0
1205 RESTORE 1160
1210 FOR I=1 TO 4
1220 READ M$
1225 LET RMOV=8500+500*SGN (VAL
M$)
1230 GO SUB EMOV
1240 GO SUB RMOV
1245 PRINT #1,AT 0,0,"PULSE UNA
TECLA PARA SEGUIR": PAUSE 0
1250 NEXT I
1260 DATA '+Y','-X","-Y","+X"
1270 PRINT #1,AT 0,0;"PULSE UNA
TECLA PARA MENU    "
1280 PAUSE 0  RETURN
1998
1999 REM  ***********************
2000 REM  **    S/R NEOTOB    **
2001 REM  ***********************
2002
2005 PAPER 1: INK 6: BORDER 1: C
LS
2010 FOR I=1 TO 9
2020 LET N(I)=I
2030 NEXT I
2040 GO SUB PRINTOB
2050 GO SUB IMOV
2060 GO SUB EMOV
2070 GO SUB RMOV
2080 GO TO 2050
2090 RETURN
2998
2999 REM  ***********************
3000 REM  **    S/R RESTOB    **
3001 REM  ***********************
3002
3005 BORDER 5: INK 1: PAPER 5  C
LS
3010 FOR I=1 TO 9
3020 LET N(I)=INT (9*RND)+1
3030 FOR J=1 TO I-1
3040 IF N(J)=N(I) THEN GO TO 302
0
3050 NEXT J
3060 NEXT I
3070 GO SUB PRINTOB
3080 GO SUB IMOV
3090 GO SUB RMOV
3100 GO SUB VTOB
3105 GO TO 3080
3110 RETURN
3998
3999 REM  ***********************
4000 REM  **     SALIDA       **
4001 REM  ***********************
4002
4005 PAPER 0: INK 7: BORDER 0: C
LS
4010 PRINT AT 3,11;"ESTA SEGURO?
";AT 5,13,"(S/N)"
4020 IF INKEY$<>"S" AND INKEY$<>
"N" THEN GO TO 4020
4030 IF INKEY$="N" THEN CLEAR :
GO TO 1
4040 PRINT AT 7,0;"ESTE PROGRAMA
 SE AUTODESTRUIRA  DENTRO DE
 MOMENTOS";
4050 FOR I=200 TO 0 STEP -1
4060 PRINT AT 8,10;I;" "
4070 NEXT I
4080 RANDOMIZE USR 0
4998
4999 REM  ***********************
5000 REM  **    S/R PRINTOB   **
5001 REM  ***********************
5002
5010 FOR I=1 TO 9
5020 CIRCLE 8*C(I)+4,175-8*F(I)-
4,20
5030 CIRCLE 8*C(I)+4,175-8*F(I)-
4,24
5040 NEXT I
5045 PRINT AT 0,13;"X";AT 18,28;
"Y",AT 21,4,"Z"
5050 FOR I=1 TO 9
5060 PRINT INK 9,AT F(I),C(I),N(
I)
5070 NEXT I
5080 RETURN
5998
5999 REM  ***********************
6000 REM  **     S/R IMOV     **
6001 REM  ***********************
6002
6010 INPUT "MOVIMIENTO? (M PARA
MENU) "; LINE M$
6020 IF M$="M" THEN CLEAR  GO T
O 1
6030 IF LEN M$<>2 THEN GO TO 601
0
6040 IF (M$(1)<>"+" AND M$(1)<>"
-") THEN GO TO 6010
6045 IF M$(2)<>"X" AND M$(2)<>"Y
" AND M$(2)<>"Z" THEN GO TO 6010
6050 LET RMOV=8500+500*SGN (VAL
M$)
6060 RETURN
6998
6999 REM  ***********************
7000 REM  **     S/R EMOV     **
7001 REM  ***********************
7002
7010 LET F1=F1+1  IF F1=21 THEN
LET F1=0  LET C1=C1+3
7020 PRINT INK 9;AT F1,C1;M$
7030 RETURN
7998
7999 REM  ***********************
8000 REM  **     S R RMOV-    **
8001 REM  ***********************
8002
8010 LET J=VAL M$(2)
8020 LET K=(J+3)-INT ((J+3)/9)*9
8030 LET L=N(K)  LET N(K)=N(J+2)
: LET N(J+2)=N(J+1): LET N(J+1)=
N(J)  LET N(J)=L
8040 FOR M-J TO J+2
8050 PRINT INK 9,AT F(M),C(M),N(
M)
8060 NEXT M
8070 PRINT INK 9;AT F(K),C(K);N(
K)
8080 RETURN
8998
8999 REM  ***********************
9000 REM  **     S/R RMOV+    **
9001 REM  ***********************
9002
9010 LET J=VAL M$(2)+3
9030 LET K=J-9*(J=10)
9040 LET L=N(J-3)  LET N(J-3)=N(
J-2)  LET N(J-2)=N(J-1): LET N(J
-1)=N(K)  LET N(K)=L
9050 FOR M=J-1 TO J-3 STEP -1
9060 PRINT INK 9;AT F(M),C(M);N(
M)
9070 NEXT M
9080 PRINT INK 9,AT F(K),C(K);N(
K)
9090 RETURN
9498
9499 REM  ***********************
9500 REM  **     S/R VTOB     **
9501 REM  ***********************
9502
9510 FOR I=1 TO 9
9520 IF N(I)<>I THEN RETURN
9530 NEXT I
9540 PRINT #1. 'LO HA LOGRADO''
'"PULSE M PARA MENU"
9550 IF INKEY$<>"M" THEN GO TO 9
550
9560 CLEAR : GO TO 1
```



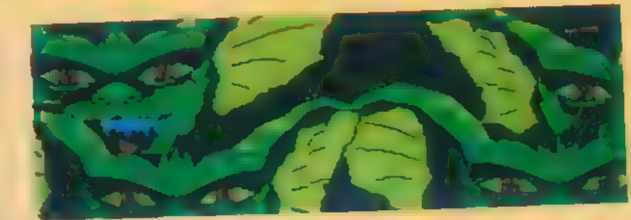
Breves notas sobre cómo exterminar a los Gremlins (Capítulo VI y último)

por Billy Puertas

Estoy en el departamento de ferretería de los grandes almacenes. Cojo el taladrador eléctrico del panel de herramientas. Voy hacia el Este hasta que aparecen de nuevo los malditos monstruos. Empujo el botón y huyen asustados a través de la única salida disponible de los conductos de ventilación. Sueldo la chapa para atraparlos en el interior. Cierro la válvula del gas. Enchufo la máquina a la red eléctrica y taladro la chapa. Cojo la sierra para metales y corto el tubo. Miro hacia el mostrador y por fin, encuentro lo que busco: la cinta adhesiva. Suelto las chapas que me quedaban en el suelo y cojo la cinta. Con ella puedo unir el tubo con la bombona de gas del soplete. Introduzco el tubo en el agujero practicado en la rejilla y abro la válvula de gas. De esta forma consigo inundar de gas todos los conductos de ventilación donde se encuentran los Gremlins. Escapo hacia el Este y luego al Sur. Salgo a la calle y suelto todos los objetos: Soplete, Máquina, Sierra y Encendedor.



En el departamento de ferretería se encuentra el providencial taladro.



Los Gremlins han sido exterminados. Por fin la ciudad puede dormir tranquila.

Espero, dándome un paseo por la ciudad hasta que estalla la bombona de gas...

Espero, dándome un paseo por la ciudad hasta que estalla la bombona de gas. Todos los Gremlins mueren víctimas de la explosión. ¡Pero que veo! ¡El maldito Stripe logró escapar! Parece que se dirige a la piscina con la intención de multiplicarse. Seguramente no sabe que está vacía. He de seguirle, ir a la piscina y cogerle. Está amaneciendo y es el momento para acabar con él. ¡Ya lo tengo! Subo hasta arriba y salgo, por el Este hacia la calle. Stripe queda fulminado por el influjo de los potentes rayos del sol.

¡Por fin la ciudad puede respirar tranquila!

ROM y RAM

Quisiera que me informasen qué es lo que significan los ROM y RAM, ¿cuántos tiene de ambos el 48K?

Antonio CABEZAS - Barcelona

☐ ROM significa «Read Only Memory» (Memoria de sólo lectura), se trata de una memoria que sólo se puede leer, pero no es posible escribir en ella. El Spectrum tiene 16K de ROM (1K son 1Ø24 bytes) que contiene el Sistema Operativo del ordenador y el intérprete de Basic.

RAM significa «Random Access Memory» (Memoria de acceso aleatorio), en la RAM se puede tanto leer como escribir. El Spectrum 48K tiene, como su nombre indica, 48K de RAM. Junto con los 16 ROM hacen un total de 64K que es la máxima cantidad de memoria direccionable por el microprocesador Z-80.

Assembler y Código Máqina

¿El Código Máquina y el lenguaje ensamblador es lo mismo? ¿Qué es o para qué sirve un programa ensamblador? ¿y un desensamblador? ¿Qué es un monitor de Código Máquina?

Quiero aprender Código Máquina y me gustaría que me recomendaseis algún libro para principiantes.

Asier BURGALETA - Donostia

☐ El Código Máquina es el conjunto de números que, almacenados en posiciones de memoria, le indican al microprocesador lo que tiene que hacer según los va leyendo. El lenguaje ensamblador o Assembler, es la traducción de estos códigos a unas palabras simbólicas, fácilmente recordables por los humanos.

Un ensamblador es un programa que realiza la traducción de Assembler a Código Máquina, un desensamblador hace exactamente lo contrario. Un monitor es un programa que facilita la depuración de una rutina escrita en C/M.

Si desea aprender Código Máquina, lo primero que le recomendamos es paciencia y muchas ganas de estudiar, aunque los resultados le valdrán la pena. Puede seguir el CURSO de CODIGO MAQUINA específico para el Spectrum que iniciamos en el número 42 de MICROHOBBY.

Prolongador de Slot

He adquirido una impresora GP-50S de SEIKOSHA, y el cable de conexión al ordenador me resultaba corto, por lo que he construido un prolongador con dos conectores en los extremos.

La carga de los programas se efectúa normalmente, pero al terminar la carga, o se borran o no se pueden ejecutar. Mi pregunta es si debido al prolongador hay pérdidas o es por otra cosa lo que me ocurre.

Sisforiano FLORES - Cádiz

☐ En primer lugar, es conveniente aislar el problema. Quite el prolongador y mire si la impresora sola le produce el mismo efecto, si no es así, ya puede estar seguro de que el problema es del prolongador.

Revise bien las conexiones porque puede estar corto-circuitando dos líneas de algún bus, y sustituya el mazo de cables por una cinta de varios conductores, para evitar el acoplamiento capacitivo entre líneas.

Sobre el Sistema Operativo

¿A qué dirección salta el microprocesador cuando introducimos un comando directo?

¿Qué significado tienen los bytes de la ROM situados desde la dirección 517 a la 653?

¿Cómo se usan los indicadores m, p, pe, po del Z-80?

José M. GOMEZ - Murcia

☐ Cuando introducimos un comando directo, el microprocesador no salta a ninguna dirección fija, depende del comando que hayamos introducido. En principio, el intérprete entra en la zona de edición a comprobar la sintaxis, si es correcta, comprueba si la línea empieza por un número, si es así, la almacena; si no, pasa a ejecutarla llamando a una gran cantidad de subrutinas en función de los comandos que incluya.

Los bytes situados entre las direcciones 517 y 653 constituyen las tablas que utiliza el Sistema Operativo para decodificar el teclado en función de los distintos modos posibles (S/S, EXT., K, etc.).

Los indicadores del Z-80 no se denominan como usted nos indica, sino: C, N, P/V, H, Z y S que corresponden respectivamente a: Acarreo, Suma/Resta, Paridad/Desbordamiento, Semiacarreo, Cero y Signo. El Z-80 los utiliza para tomar decisiones en función del resultado de los cálculos realizados se utilizan principalmente, en los saltos condicionales.

Código Máquina

Me gustaría que publicaseis una serie de artículos dedicados a la iniciación al Código Máquina para aquellos que, como yo, queramos iniciarnos en él, pero le tengamos demasiado respeto.

Jesús M. SESAR - Santiago de Compostela

☐ Como probablemente haya tenido ocasión de comprobar, sus deseos han sido satisfechos a partir del número 42 de MICROHOBBY.

Periféricos

¿Qué es un periférico y para qué vale?

¿Qué es el PIO o PI/O, para qué se usa y cuánto vale?

José L. ALONSO - Alcobendas

☐ Un periférico es cualquier dispositivo de un ordenador que no es el microprocesador ni la memoria. Su utilidad es la de comunicar al ordenador con el mundo exterior. En el Spectrum son periféricos: el teclado, la pantalla, el cassette, el microdrive, la impresora, el joystick, etc.

Un PIO o PI/O, siglas de «Paralel Input/Output» (Entrada/Salida en Paralelo), es un circuito integrado que se encarga de gestionar la comunicación de un microprocesador con el exterior. Los hay de distintos tipos dependiendo del microprocesador con el que vayan a trabajar.

Ultimate

He visto en muchas revistas cantidad de reportajes sobre casas de software, pero no he visto ninguno sobre ULTIMATE. ¿No les parece un buen tema para un reportaje de su revista?

Antonio MARTINEZ - Madrid

☐ Los programadores de ULTIMATE permanecen en el anonimato por razones de competencia comercial y hasta el momento, que sepamos, no han concedido ninguna entrevista a la prensa.

Rótulos gigantes

Tengo un Spectrum 48K, y quiero saber si existe alguna dirección de memoria, orden o programa que me permita alterar el tamaño y la separación entre letras.

Eugenio ROMERO - Ciudad Real

☐ En el número 8, página 16 de MICROHOBBY, publicábamos un artículo sobre la forma de utilizar la rutina de rótulos de la cinta «HORIZONTES» de demostración que acompaña al Spectrum, esta rutina es bastante potente si tenemos en cuenta la poca memoria que ocupa (277 bytes).



DE OCASION

● DESEARIA contactar con gente de cualquier punto de Cataluña que sepan programar en Spectrum. Escribir a Xavi Betrán. Isard, 6, 5.º. Barcelona 08031.

● VENDO Spectrum 48 K, en buen estado, libro de instrucciones y bibliografía diversa. Todo por 30.000 ptas. Además regalo un Currah Uspeechy y un teclado profesional. También vendo consola Atari de vídeo-juegos, 18 cartuchos, 4 joystick por 25 000 ptas. Interesados escribir a Pedro González Medina. Juan Ponce Pérez, Las Palmas de Gran Canaria 35017.

● VENDO ordenador ZX-81. El precio incluye manual de instrucciones en castellano, fuente de alimentación, cables para el cassette y para TV. Llamar el Tel. (94)4326867 de Bilbao. Preguntar pro Joaquín Lavín.

● VENDO ZX Spectrum 16 K, con todos los accesorios (manual, cables, fuente, cinta de demostración Horizontes) y algunas revistas de regalo. Todo en perfecto estado de uso y por el precio de 25.000 ptas. Interesados contactar con Pedro Fernández Sierra. Del Mar, 33. Soller (Mallorca).

● INTERCAMBIO todo tipo de información para Spectrum. También vendo copión turbo original importado de Inglaterra por 3.000 ptas. Incluyo en la misma cinta dos copiones más. Escribir a Juan Leonardo de Haro. Beas, 4. Huelva 21006. Tel. (955)224455.

● ME GUSTARIA mantener correspondencia con personas de España que tengan microcomputadora. Mi dirección es la siguiente: Angel Fabián Dángela. San Lorenzo, 2647. Distrito 2. Rosario (CP-2000). Santa Fe (Argentina).

● CAMBIO una unidad de microdrive (ZX Interface I y ZX Microdrive). Incluyo también dos cartuchos en blanco de Sinclair, además de cables de conexión. para más información dirigirse al Tel. (91)7397362. Preguntar por José Gisbert. Madrid.

● DESEARIA que algún lector me mandase el mapa del juego «The Lords of Midnight», o bien las instrucciones del «Valhalla». Pagaré fotocopias. Llamar al Tel. (928)252967, para más información o bien dirigirse a la siguiente dirección: Antonio Marcos López. Doña Perfecta, 20, 4.º derecha. Las Palmas de Gran Canaria.

● VENDO vídeo-juego Philips por 20.000 ptas., escribir o llamar al Tel. 315469. Luis José Macías. Baños, 14. Mérida (Badajoz).

● VENDO cassette Gold King, especial para Spectrum, alta sensibilidad, con salida Ear, Mic y altavoz. Precio: 5.000 ptas. (negociables). Importe total en venta: 8.700 ptas. Interesados llamar al Tel. (91)7155044. Madrid (a partir de las 5, preguntar por Jorge Arias).

● VENDO Spectrum 16 K, un año, garantía, manual en castellano. Precio: 21.000 ptas. Ofertas a José Luis al Tel. (985)345437 (de 1 a 4). Gijón (Asturias).

● VENDO por 25.000 ptas. Microdrive con su correspondiente Interface-1, a estrenar, ya que me correspondió en un sorteo y no dispongo de ordenador Spectrum. Interesados en la oferta escribir a Javier Iturriaga. Siervas de Jesús, 27. Haro (La Rioja).

● COMPRO O CAMBIO las instrucciones en castellano del compilador Pascal para Spectrum 48 K de Hisoft, HP4S o HP4T. José M. Perona. Tel. (968)760371. Murcia.

● INTERESADO en formar un club de usuarios del Spectrum en Cartagena para intercambio de ideas, información, libros y sobre todo para hacer nuestros propios programas, escriban a: Salvador Martínez Otón. Duque Severiano, 6, 2.º Cartagena (Murcia).

● VENDERIA lote compuesto por vídeo-juegos Philips Videopac Plus G7400, dos cartuchos ROM de juegos máxima resolución y otro también ROM de resolución normal. Es convertible a ordenador de bastante memoria mediante la compra de un periférico. Comprado en enero-85. Garantizo ahorro de 13.000 ptas. y obsequio con 3 libros. Interesados llamar al Tel. (982)218979 (9 a 11 noche).

● VENDO Spectrum Plus, nuevo a estrenar, manual, caja, cables, etc. Precio especial: 32.000 ptas. Escribir a la siguiente dirección: Antonio Polo. Gran Vía Carlos III, 127, 7.º Barcelona 08034.

● CLUB ZX poseemos una amplia gama de cassette, libros sobre Basic, revistas, etc. Interesados pueden escribir a José Antonio Méndez Ríos. Príncipe Felipe, 4, 3.º Ceuta. Tel. 518829.

● DESEARIA que algún lector me enviara por correo las instrucciones de los programas siguientes: «Monitor», de Cristal, «Zeuss Assembler», «Beta Basic». Pagaría gastos de envío y fotocopias. Manuel Sayas. Arturo Soria, 301, 6.º D. Madrid.

● VENDO Spectrum Plus 48 K (tranformador, instrucciones castellano, cinta) por el precio de 43.000 ptas. Regalo libros, revistas. También vendo cassette Philips por 10.000 ptas. Asi mismo vendo Organo Electrónico Casiotone 501, con regalo de más de 100 partituras, por el precio de 60.000 ptas. Interesados llamar al Tel. 2233770 de Barcelona. Alejandro.

● VENDO Spectrum 16 K, con cables, manuales en castellano y cinta Horizontes de demostración. Precio: 31.000 ptas. Contactar con Alfonso Pastor. Plaza P. Valenciano, 4 Rocafort (Valencia). Tel. (96)1310144.

● VENDO Spectrum 48 K, más impresora ZX, 2 rollos de papel, 23 revistas. Todo por 40.000 ptas. Tel. 832077. Cantabria (Santander). Llamar sólo mañanas. Alberto.

● DESEARIA contactar con chicas que tengan Spectrum (48 K o Plus) de Barcelona. Escribir a Alfonso Loma Gallego. Corneliá, 10, 3.º 4.ª Barcelona.

● VENDO interruptor ON/OFF para Spectrum por sólo 850 ptas. También vendo algunas revistas a la mitad de precio. Interesados llamar al Tel. (954)663404. Sevilla.

● VENDO ZX Spectrum 16 K, con fuente de alimentación, conectores y cinta de demostración. Precio: 22.000 ptas. (negociables). Regalaría revistas ZX y Todospectrum. Llamar al Tel. (976)517156. Preguntar por Pedro Antonio.

● CAMBIO/VENDO equipo de Ski, poco uso. Pedir inforamción por Tel. Se vende por 35.000 ptas. O lo cambio por un ordenador ZX Spectrum 48 K, cables, manual, fuente y demás accesorios como la cinta de demostración, libros de instrucciones, etc. Interesados llamar al Tel. (987)223760 de León. Preguntar por Manolo.

● VENDO ZX Spectrum 48 K RAM por el precio de 28.000 ptas., incluyo el libro: «¿Qué es para que sirve?».

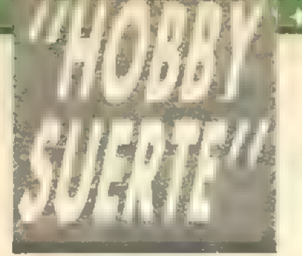
● VENDO ordenador ZX Spectrum en perfecto estado con todos sus accesorios también va incluido la unidad de cassette, revistas, libro de programación BASIC BASICO. Precio 50 000 ptas. Interesados llamar al Tel. (924)530354 (tardes de 6 a 9).

● VENDO Spectrum 48 K, a estrenar, con manual en español y cinta de demostración, opcional también cassette reproductor para el ordenador. Precio a convenír. Tel. 73446802. Madrid. Preguntar por José Luis.

● INTERCAMBIO información, trucos, ideas, para Spectrum. Escribe a Rafael Antón Sánchez. Apdo. Correos 451. Elche (Alicante)

● CAMBIARIA algunos de los siguientes artículos, a elegir, por aparato Stereofónico, de doble platina, Philips: Interface programable y de sonido (Indescomp). Joystick (Gran Capitán). Consola de vídeo-juegos G-7000 (Philips). Llamar al Tel. 8507789 de Madrid. Preguntar por Oscar.

● VENDO Joystick más Interface Kempson por 5.000 ptas. Si te interesa escribe a Julián Seguen García. Serradilla, 28, 2.º D. Madrid 28044. Tel. 7059824.



¡250.000 pts. en premios cada semana!

ESTE NUMERO PUEDE SIGNIFICAR UN FABULOSO REGALO PARA TI.

33908722

¡Consulta a tu Spectrum!

Cada semana, Microhobby regala 70 premios entre sus lectores. La clave del premio es el número que figura en este cupón, en la esquina superior derecha.

Para saber si el número de tu ejemplar está premiado, debes introducirlo en tu Spectrum, utilizando para ello el programa «Hobby-Suerte». La cassette con este programa se ha entregado a los lectores junto con el número 15 de Microhobby Semanal. Si no posees esta cinta, puedes pedir una copia a un amigo o por carta a **Hobby Press, S.A., Apartado n.° 54062. Madrid,** incluyendo dentro del sobre 180 pts. en 3 sellos de Correos de 60 pts. cada uno. Este programa sirve para leer todos los números durante las **50 semanas** que dure este Concurso.

(Bases en el reverso)

Premios semanales

■ Primera Categoría

Un **Spectrum 48 k.** (o un Microdrive y un Interface 1, a elegir por el interesado).

■ Segunda Categoría

Una **Impresora GP 50 de Seikosha,** especialmente diseñada para Spectrum (2 premios).

■ Tercera Categoría

Un **Joystick con su interface** (3 premios).

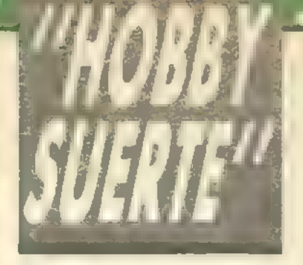
■ Cuarta Categoría

Una **Suscripción a Microhobby Semanal** por un año (50 números. Si el lector premiado ya es suscriptor, podrá optar por prolongar su suscripción anual o un premio de Quinta Categoría) (14 premios).

■ Quinta Categoría

Una **Cinta de Programa,** a elegir entre un variado surtido de juegos, utilidades, etcétera (50 premios).

Hobby Press, S.A. garantiza que cada semana introduce al azar, entre todos los ejemplares que componen la edición, **setenta cupones** correspondientes a los premios aquí citados



Instrucciones para concursar

Una vez introducido en memoria el programa **«Hobby-Suerte»,** aparecerá en la pantalla la clásica máquina «tragaperras» de frutas.
A continuación, debes teclear el número que figura en esta tarjeta. Al pulsar «Enter», la «máquina» se pone en marcha y te hace saber si has sido agraciado con uno de los 70 premios semanales.

Muy importante: Puede ocurrir que, al introducir en el programa números al azar, alguno de ellos corresponda casualmente a un premio. También es posible, con los suficientes conocimientos de Basic, alterar el programa para que un número determinado aparezca como premiado.

Por todo ello, debemos aclarar los siguientes aspectos:

1. El único justificante para reclamar un premio determinado es la posesión del cupón con el número impreso en él.

2. Todos los números susceptibles de dar premio están registrados ante Notario.

3. Hobby Press, S.A. no se hace responsable de ningún otro cupón que no corresponda a los números previamente registrados. Tampoco se atenderán reclamaciones verbales que no vengan acompañadas por la posesión del cupón con el número premiado.

4. Cualquier lector puede solicitar de esta Editorial la comprobación de la entrega de los Premios semanales.

5. Hobby Press, S.A. se reserva el derecho a resolver según su criterio cualquier cuestión no prevista en las Bases de este Concurso.

6. La reclamación de cualquier Premio de este Concurso caduca el día 30 de Junio de 1986

COMUNICACION DE PREMIO

(Enviar rellenado con letra clara y en sobre cerrado)

Nombre .... Edad ....
Apellidos ....
Domicilio .... Teléfono ....
Ciudad .... C P .... Provincia ....
Categoría del Premio Obtenido .... Número de Microhobby ....

Si consideras que tu cupón tiene premio, fotocópialo como medida de seguridad y envíalo por **Correo Certificado** a Hobby Press, S.A. Apartado 54.062 de Madrid. Por favor, anticípanos todos estos datos por teléfono llamando al (91) 654 32 11. En este mismo número atenderemos cualquier consulta o duda sobre las Bases o la mecánica de «Hobby Suerte»

Envía este cupón por correo certificado a HOBBY PRESS. Apt. 54.062. Madrid

¡¡FABULOSO... HIPERSPORTS!!

1 ☐ **SI**, deseo suscribirme a la Revista **Microhobby Semanal** durante un año (50 números), al precio actual de **4.250** pesetas, y recibir como regalo, sin gastos de envio por mi parte, una cinta original del programa «Hypersports», valorada en 2.100 pesetas.

(Debido a lo excepcional de esta oferta, y hasta el 30 de septiembre, nos vemos obligados a suspender las modalidades de pago mediante reembolso o Tarjeta de Crédito. Por favor, envie un talón en un sobre cerrado junto con este cupón o remita un giro postal. Gracias)

2 ☐ **SI**, deseo recibir la(s) **Cinta(s) de Programas** que indico a continuación. Cada una de estas cintas lleva grabados los programas publicados por Microhobby Semanal durante cuatro numeros consecutivos (1 al 4, 5 al 8, 9 al 12 etcétera) y su precio es de 550 pesetas, más 75 ptas. por gastos de envio cada pedido (no por cada cinta).

Números del ___ al ___, (inclusive) Números del ___ al ___, (inclusive)
Numeros del ___ al ___, (inclusive) Números del ___ al ___, (inclusive)
Números del ___ al ___, (inclusive) Números del ___ al ___, (inclusive)

(Para recibir numeros atrasados de Revistas o estuches de encuadernación solicitelos mediante carta o tarjeta postal o por teléfono a los nums. (91) 733 50 12 y (91) 733 50 16. El precio de cada ejemplar es el mismo que figure en la portada y los estuches 495 ptas. cada uno. Los pedidos contra reembolso se incrementarán con 75 ptas. de gastos por cada envio)

OFERTAS VALIDAS SOLO PARA ESPAÑA

NOMBRE ___ EDAD ___
APELLIDOS ___
DOMICILIO ___
CIUDAD ___ PROVINCIA ___
C. POSTAL ___ TELEFONO ___ PROFESION ___

Marco con una (X) en el casillero correspondiente la forma de pago que más me conviene

☐ Talón bancario adjunto a nombre de HOBBY PRESS, S.A. ☐ Giro Postal N. ___ [ ] Contra reembolso del prime. N

TARJETA DE CREDITO [ ] VISA N. ___ ☐ MASTER CHARGE N. ___

Fecha de caducidad de la tarjeta ___ Firma

(cortar por la linea de trazos)

Franqueo
Postal

HOBBY PRESS, S.A.

Apartado de Correos

n.º **54.062** (Apartados Altos)

28080 MADRID

Ensamblador



Un ensamblador es una herramienta de software (un programa), diseñado para simplificar las tareas que conlleva escribir los programas en código máquina, bien en binario o en hexadecimal.

El lenguaje ensamblador es una serie de nombres simbólicos (mnemónicos) de operación, fácilmente comprensibles, que se corresponden con las microinstrucciones de la CPU (Unidad Central de Proceso), lo cual obliga al programador de lenguaje ensamblador a conocer detalladamente cada una de las operaciones que ésta realiza.

Para usar el lenguaje ensamblador necesitamos un fichero de código fuente, que es una lista de líneas de texto, que deben cumplir las siguientes exigencias:

1. **Número de línea,** por cuyo orden son colocadas y ensambladas, a semejanza del Basic.

2. **Campo de etiqueta,** referencia necesaria para que el ensamblador desarrolle el flujo de programa deseado, en saltos u otras instrucciones que manejen direcciones.

Código fuente
Código objeto
Código máquina
Líneas de ensamblador
Campos
Ensamblaje en 2 pasos

3. **Campo de código de operación** (mnemónico), es opcional, y puede contener en lugar del código una directiva de ensamblador (pseudoinstrucción).

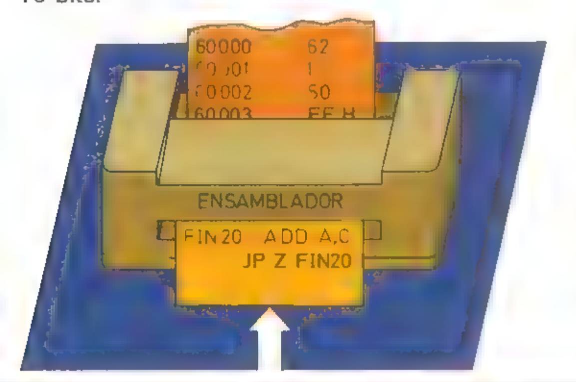
4. **Campo de operando,** también opcional, respetando la estructura del código mnemónico, puede tener ningún, uno o dos operandos, en este último caso deben ir separados por coma, y siempre que sean numéricos, pueden ser sustituidos por una expresión simbólica (con etiquetas).

5. **Campo de comentario,** opcional, de ayuda para entender mejor los programas, debe ir precedido de un punto y coma.

Todos los campos de una línea deben estar separados al menos por un espacio, siendo acon-

sejable el empleo de tabulaciones, para que queden alineados por columnas, que contribuye al mejor entendimiento del programa.

• Una expresión numérica en lenguaje ensamblador es una combinación de números, símbolos y operadores, respetando las reglas algebraicas, donde cada elemento de la expresión es un término, y el resultado debe estar acorde con el operando a que sustituye, en su rango, de 8 a 16 bits.



Normalmente una expresión numérica debe poder admitir números en cualesquiera de las bases corrientemente utilizadas en lenguaje ensamblador, o sea, binario, octal, decimal o hexadecimal.

Una vez tenemos el código fuente, podemos ensamblarlo, en dos pasos, para producir el código objeto.

• En ensambladores más potentes, normalmente con ordenadores de mayor tamaño, el fichero de código objeto se combina con otros ficheros para generar el código máquina, y en ensambladores más sencillos, este constituye directamente el propio código máquina, que es el ejecutable por la CPU.

En el primer paso se comprueban errores de sintaxis, errores de organización de memoria, y se calculan el espacio necesario y los desplazamientos de las direcciones relativas.

En el segundo paso, si no ha habido errores, se cumplimenta el código objeto, chequeando que los valores de los operandos estén en su rango, y las etiquetas estén en su lugar correcto (no haya etiquetas repetidas o inexistentes).



LD (IY+d),n

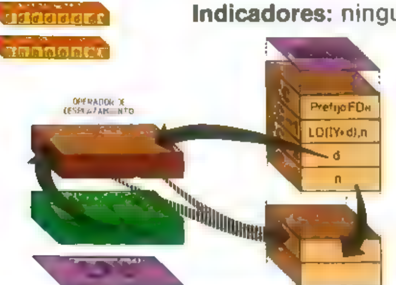
El número de 8 bits n es transferido a la dirección de memoria especificada por la suma del contenido del par IY y el desplazamiento d (d es un número de 8 bits en complemento a 2).

Mnemónico: LD   **Operandos:** (IY+d),n

Formato binario:

Ciclos: 5
Estados: 19 (4,4,3,5,3)

Indicadores: ninguno



Instr.	Hex.	Dec.
LD (IY+d),n	FD,36,d,n	253,54,d,n
LD (IY+d),A	FD,77,d	253,119,d
LD (IY+d),B	FD,70,d	253,112,d
LD (IY+d),C	FD,71,d	253,113,d
LD (IY+d),D	FD,72,d	253,114,d
LD (IY+d),E	FD,73,d	253,115,d
LD (IY+d),H	FD,74,d	253,116,d
LD (IY+d),L	FD,75,d	253,117,d
LD A,(IY+d)	FD,7E,d	253,126,d
LD B,(IY+d)	FD,46,d	253,70,d
LD C,(IY+d)	FD,4E,d	253,78,d
LD D,(IY+d)	FD,56,d	253,86,d
LD E,(IY+d)	FD,5E,d	253,94,d
LD H,(IY+d)	FD,66,d	253,102,d
LD L,(IY+d)	FD,6E,d	253,110,d

LD (IY+d),r

El contenido de cualquier registro r es transferido a la dirección de memoria especificada por la suma del contenido del par IY y el desplazamiento d (d es un número de 8 bits en complemento a 2).

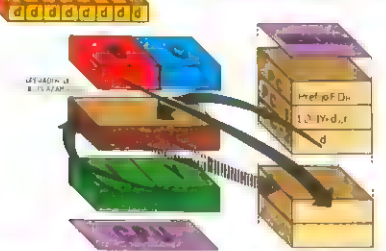
Mnemónico: LD **Operandos:** (IY+d),r

Formato binario:

1	1	1	1	1	1	0	1
0	1	1	1	0	r	r	r
d	d	d	d	d	d	d	d

Ciclos: 5
Estados: 19 (4,4,3,5,3)

Indicadores: ninguno



LD r,(IY+d)

El contenido de la dirección de memoria especificada por la suma del contenido del par IY y el desplazamiento d (d es un número de 8 bits en complemento a 2), es transferido a cualquier registro r.

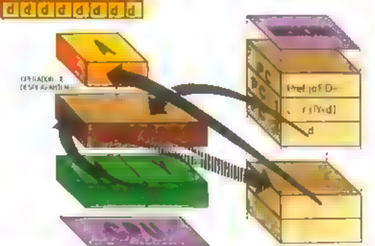
Mnemónico: LD **Operandos:** r,(IY+d)

Formato binario:

1	1	1	1	1	1	0	1
0	1	r	r	r	1	1	0
d	d	d	d	d	d	d	d

Ciclos: 5
Estados: 19 (4,4,3,5,3)

Indicadores: ninguno





LD dd,nn

El número nn de 2 bytes, es transferido al par de registros especificado por el operando dd.

Nnemónico: LD **Operandos:** dd,nn

Formato binario:



Ciclos: 3
Estados: 10 (4,3,3)

Indicadores: ninguno

Instr.	Hex.	Dec.
LD BC,nn	01,n,n	1,n,n
LD DE,nn	11,n,n	17,n,n
LD HL,nn	21,n,n	33,n,n
LD SP,nn	31,n,n	49,n,n
LD IX,nn	DD,21,n,n	221,33,n,n
LD IY,nn	FD,21,n,n	253,33,n,n

Ejemplo:

Después de ejecutar la instrucción
LD BC,4000H
resultará que el par BC contiene 4000H.

El código del par dd, para la construcción del código binario de la instrucción es:

BC	00
DE	01
HL	10
SP	11

LD IX,nn

El número nn de 2 bytes, es transferido al par IX.

Mnemónico: LD **Operandos:** IX,nn

Formato binario:

11011101
00100001
nnnnnnnn
nnnnnnnn

Ciclos: 4
Estados: 14 (4,4,3,3)

Indicadores: ninguno

LD IY,nn

El número nn de 2 bytes, es transferido al par IY.

Mnemónico: LD **Operandos:** IY,nn

Formato binario:

11111101
00100001
nnnnnnnn
nnnnnnnn

Ciclos: 4
Estados: 14 (4,4,3,3)

Indicadores: ninguno

Giro Horizontal

Esta rutina realiza un giro de la pantalla tomando como eje una línea vertical situada en el centro de la misma.

La forma de llamada es la usual, es decir:

RANDOMIZE USR n

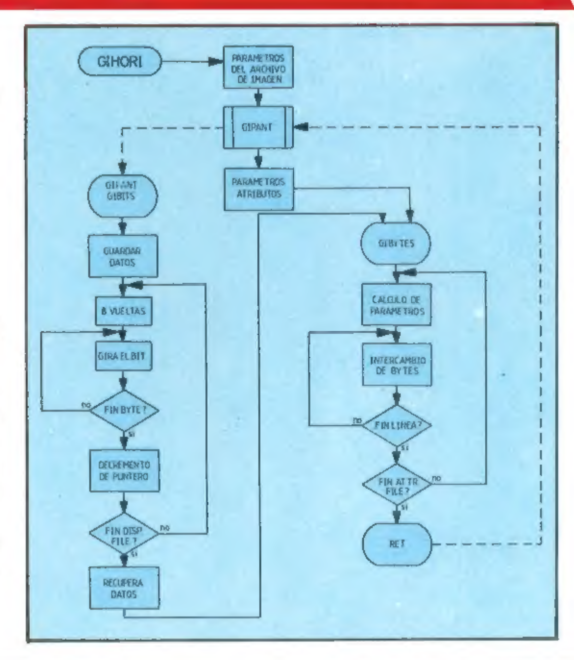
siendo n la dirección a partir de la cual se ha situado la rutina (es relocatable).

Funcionamiento:

Utiliza la subrutina llamada GIPANT compuesta a su vez por otras dos subrutinas cuyos nombres son GIBITS y GBYTES.

La primera parte de la rutina trabaja en el fichero de pantalla, invirtiendo cada una de las 8 líneas de puntos de cada carácter sobre sí mismas, bit a bit (GIBITS), trasladándolas después a su dirección definitiva, al otro lado de la pantalla (GBYTES).

Por último intercambiará los atributos de los caracteres (CALL GBYTES), localizando su dirección en el fichero de atributos.



```
10 ; *  GIRO HORIZONTAL  *
20        ORG    60000     ;RUTINA NO REUBICABLE
30 START  LD     HL,16384  ;Comienzo de pantalla
40        LD     DE,#1800  ;Long. DISPLAY FILE
50        CALL   GIPANT    ;Gira DISPLAY FILE
60        LD     DE,#300   ;Long. archivo atrib.
70        JR     GBYTES    ;Gira arch. atributos
80 GIPANT
90 GIBITS PUSH   HL        ;Com. de pantalla
100       PUSH   DE        ;Long. arch. atribut.
110 BUCLE1 LD    B,8       ;No. de bits por byte
120       LD     A,(HL)
130 BUCLE2 RLA   ;         ;Extrae bit
140       RR     (HL)      ;Guarda bit
150       DJNZ   BUCLE2
160       INC    HL        ;Puntero
170       DEC    DE        ;Longitud
180       LD     A,E
190       OR     D
200       JR     NZ,BUCLE1;6144 vueltas
210       POP    DE        ;Recupera longitud
220       POP    HL        ;Recupera comienzo
230       LD     C,32      ;Anchura de linea
240 GBYTES PUSH  HL        ;Puntero
250       PUSH   DE        ;Longitud
260       PUSH   BC        ;Anchura
270       LD     E,L
280       LD     D,H       ;Transfiere HL a DE
290       ADD    HL,BC     ;Incrementa anchura
300       DEC    .HL       ;Puntero A
310       SRL    C         ;C/2
320 BUCLE3 LD    A,(HL)
330       LD     B,A       ;Cambia
340       LD     A,(DE)    ;contenido DE
350       LD     (HL),A    ;por cotenido de HL
360       LD     A,B
370       LD     (DE),A
380       DEC    HL        ;Puntero A
390       INC    DE        ;Puntero B
400       DEC    C         ;Ancho divid. entre 2
410       JR     NZ,BUCLE3
420       POP    BC        ;Ancho
430       POP    HL        ;Puntero
440       OR     A         ;Carry a 0
450       SBC    HL,BC     ;Resta ancho
460       EX     DE,HL     ;Lo transfiere a DE
470       POP    HL        ;Puntero
480       ADD    HL,BC     ;Suma ancho
490       LD     A,D
500       OR     E         ;Continua el bucle
510       JR     NZ,GBYTES;  si DE<>0
520       RET    ;         ;Si DE=0 fin
```

```
10 DATA "21 00 40 11 00 18 CD 6E",453
20 DATA "EA 11 00 03 18 14 E5 D5",740
30 DATA "06 08 7E 17 CB 1E 10 FB",663
40 DATA "23 1B 7B B2 20 F2 D1 E1",1071
50 DATA "0E 20 E5 D5 C5 5D 54 09",871
60 DATA "2B CB 39 7E 47 1A 77 78",765
70 DATA "12 2B 13 0D 20 F5 C1 E1",788
80 DATA "B7 ED 42 EB E1 09 7A B3",1256
90 DATA "20 E0 C9                ",457
```